NASA TECHNICAL NOTE

NASA TN D-7831

NASA TN D-7831

A FORTRAN PROGRAM FOR DETERMINING AIRCRAFT STABILITY AND CONTROL DERIVATIVES FROM FLIGHT DATA

Richard E. Maine and Kenneth W. Iliff

Flight Research Center
Edwards, Calif. 93523

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1975

1. Report No. TN D-7831	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A FORTRAN PROGRAM FOR DETERMINING AIRCRAFT STABILITY AND CONTROL DERIVATIVES FROM FLIGHT DATA		5. Report Date April 1975 6. Performing Organization Code
7. Author(s) Richard E. Maine and Kenneth W. Iliff		8. Performing Organization Report No. H-856 10. Work Unit No. 501-26-01
9. Performing Organization Name and Address NASA Flight Research Center P. O. Box 273 Edwards, California 93523		11. Contract or Grant No. 13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A digital computer program written in FORTRAN IV for the estimation of aircraft stability and control derivatives is presented. The program uses a maximum likelihood estimation method. Two associated programs for routine, related data handling are also included. The three programs form a package that can be used by relatively inexperienced personnel to process large amounts of data with a minimum of manpower. This package has been used to successfully analyze 1500 maneuvers on 20 aircraft. It is designed to be used without modification on as many types of computers as feasible. Program listings and sample check cases are included.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Aircraft stability and control Maximum likelihood estimator FORTRAN computer program	Unclassified - Unlimited Category: 59

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	175	$6.25

For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

A FORTRAN PROGRAM FOR DETERMINING AIRCRAFT STABILITY AND CONTROL DERIVATIVES FROM FLIGHT DATA

Richard E. Maine and Kenneth W. Iliff
Flight Research Center

INTRODUCTION

Determination of aircraft stability and control derivatives from flight data is of great importance in flight testing and control system design. Several methods have been used, but recent interest has turned toward maximum likelihood estimators. In addition to producing the "best" possible estimates as defined by some probabilistic criterion, these methods can be automated to a large extent.

Experience at the NASA Flight Research Center has shown that derivatives can be extracted with minimum effort by relatively inexperienced personnel using maximum likelihood estimators. Others have had some difficulty, perhaps partially due to inadequately designed programs. A production version of a maximum likelihood estimation program has been developed and used at the Flight Research Center to determine aircraft stability and control derivatives from large amounts of flight data. The program was designed to be compatible with as many types of computers as feasible and was structured to accommodate alterations easily. The program is applicable to many linear parameter estimation problems, although several of the features are intended specifically for aircraft stability and control applications. Reference 1 discusses an earlier program from which this maximum likelihood estimation program was conceptually derived.

This report presents the modified maximum likelihood estimation computer program used at the Flight Research Center for derivative extraction as well as associated programs for table lookup of initial estimates of the derivatives and for plotting results. Program listings and sample check cases for each program are included in the appendixes.

SYMBOLS

Parenthetical symbols are computer identifiers for data channels.

A stability matrix, or axial force (appendix E)

a_n (AN)	vertical acceleration, g
a_x (AX)	longitudinal acceleration, g
a_y (AY)	lateral acceleration, g
B	control matrix
C_m	dimensionless pitching-moment coefficient
C_n	dimensionless yawing-moment coefficient
C_Z	dimensionless normal-force coefficient
c	vector of unknowns
c_0	*a priori* value of c
$D1$	signal weighting matrix
$D2$	*a priori* weighting matrix
$E[\ ]$	expected value
E_k	relative error
G	observation matrix
g	acceleration of gravity, m/sec^2 (ft/sec^2)
H	observation matrix
I	identity matrix
I_X	moment of inertia about the longitudinal axis, kg-m^2 (slug-ft^2)
I_{XZ}	cross-product of inertia about the longitudinal and normal axes, kg-m^2 (slug-ft^2)
I_Z	moment of inertia about the normal axis, kg-m^2 (slug-ft^2)
i	time index
J	cost functional

L	rolling moment divided by moment of inertia about longitudinal axis, rad/sec^2
$L_0, L_{0_2}, L_{0_3}, L_{0_4}$	rolling acceleration equation biases
M	pitching moment divided by moment of inertia about lateral axis, rad/sec^2
$M_0, M_{0_2}, M_{0_3}, M_{0_4}$	pitching acceleration equation biases
N	yawing moment divided by moment of inertia about normal axis, rad/sec^2, or number of time points
$N_0, N_{0_2}, N_{0_3}, N_{0_4}$	yawing acceleration equation biases
p (P)	roll rate, deg/sec or rad/sec
q (Q)	pitch rate, deg/sec or rad/sec
$\bar{q}$	dynamic pressure, N/m^2 (lb/ft^2)
R	acceleration transformation matrix
r (R)	yaw rate, deg/sec or rad/sec
S	reference area, m^2 (ft^2)
s	auxiliary time variable, sec
T	total time, sec
t	time, sec
Δt	time interval between samples, sec
u	control vector
V	velocity, m/sec (ft/sec)
ν	variable bias vector
W	aircraft weight, N (lb)
X	longitudinal force divided by mass, m/sec^2 (ft/sec^2)
$X_0, X_{0_2}, X_{0_3}, X_{0_4}$	longitudinal acceleration equation biases

x	state vector
Y	side force divided by mass and velocity, rad/sec
$Y_0, Y_{0_2}, Y_{0_3}, Y_{0_4}$	side force equation biases
y	computed observation vector
Z	normal force divided by mass and velocity, rad/sec
$Z_0, Z_{0_2}, Z_{0_3}, Z_{0_4}$	normal force equation biases
z	measured observation vector
α (A)	angle of attack, deg or rad
β (B)	angle of sideslip, deg or rad
δ	control, deg or rad
δ_a (DA)	aileron position, deg or rad
$\delta_c, \delta_1, \delta_2$ (DC, D1, D2)	extra controls, deg or rad
δ_e (DE)	elevator position, deg or rad
δ_r (DR)	rudder position, deg or rad
η	noise vector
θ (THET)	pitch attitude, deg or rad
$\dot{\theta}_0, \dot{\theta}_{0_2}, \dot{\theta}_{0_3}, \dot{\theta}_{0_4}$	biases in Euler pitch rate equation
τ	revised time interval, sec
φ (PHI)	Euler roll attitude, deg or rad
$\dot{\varphi}_0, \dot{\varphi}_{0_2}, \dot{\varphi}_{0_3}, \dot{\varphi}_{0_4}$	biases in Euler roll rate equation

∇_c	gradient with respect to c
∇_c^2	second gradient with respect to c (Hessian matrix)
$\mathbf{O}$	null matrix

Superscript:

*	transpose

Subscripts:

$p,q,r,V,\alpha,\beta,$ $\delta_a,\delta_c,\delta_e,$ $\delta_r,\delta_1,\delta_2$	partial derivatives with respect to the subscripted variable
i,k	i^{th} and k^{th} elements of vector or matrix
L	iteration number
0	constant value

A dot over a quantity denotes the time derivative of that quantity.

PARAMETER ESTIMATION

The problem considered is: Given a set of flight time histories of an aircraft's response variables, find the values of some unknown parameters in the system equations that best represent the actual aircraft response. An intuitive mathematical approach to this problem would be to minimize the difference between the flight response and the response computed from the system equations. This difference could be defined for each response variable as the integral of the error squared. These signal errors could then be multiplied by weighting factors and summed to obtain the total response error, thereby defining an integral squared error criterion.

A mathematically more precise formulation can be made in probabilistic terms. For each possible estimate of the unknown parameters, a probability that the aircraft response time histories take on the values actually observed can be defined. The estimates should be chosen so that this probability is maximized. This process is referred to as a maximum likelihood formulation of the problem. Maximum likelihood estimators have many desirable characteristics; for example, they yield asymptotically unbiased and consistent estimates. If the measurement noise is assumed to be Gaussian, white, stationary, and uncorrelated, this formulation is equivalent to a response error formulation, in which the weightings used are the inverse of the measurement noise covariance matrix.

To mathematically describe the maximum likelihood estimator it is first necessary to define the equations of motion for the aircraft system. These equations are:

$$R\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

$$y(t) = \left[-\frac{I}{G}-\right]x(t) + \left[-\frac{\mathbf{0}}{H}-\right]u(t) + \left[-\frac{\mathbf{0}}{\nu}-\right] \tag{2}$$

$$z(t) = y(t) + \eta(t) \tag{3}$$

where

x	state vector
u	control vector
ν	bias vector
y	computed observation vector
z	measured observation vector
η	noise vector

For the aircraft problem being considered, it is convenient to separate the equations of motion into longitudinal and lateral-directional sets. The linearized longitudinal equations are:

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ q\\ V\\ \theta\end{bmatrix} = \begin{bmatrix} Z_\alpha & 1 & Z_V & -\sin(\theta)\cos(\varphi)\frac{g}{V} \\ M_\alpha & M_q & M_V & 0 \\ X_\alpha & 0 & X_V & -\cos(\theta)g \\ 0 & \cos(\varphi) & 0 & 0 \end{bmatrix}\begin{bmatrix}\alpha\\ q\\ V\\ \theta\end{bmatrix} + \begin{bmatrix} Z_{\delta_e} & Z_{\delta_c} & Z_{\delta_1} & Z_{\delta_2} & Z_0 \\ M_{\delta_e} & M_{\delta_c} & M_{\delta_1} & M_{\delta_2} & M_0 \\ X_{\delta_e} & X_{\delta_c} & X_{\delta_1} & X_{\delta_2} & X_0 \\ 0 & 0 & 0 & 0 & \dot{\theta}_0 \end{bmatrix}\begin{bmatrix}\delta_e\\ \delta_c\\ \delta_1\\ \delta_2\\ 1\end{bmatrix} \tag{4}$$

$$a_n = -\frac{V}{g}\left[\dot{\alpha} - q + \sin(\theta)\cos(\varphi)\frac{g}{V}\theta\right] + a_{n_{bias}} \tag{5}$$

The linearized lateral-directional equations are:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\frac{I_{XZ}}{I_X} & 0 \\ 0 & -\frac{I_{XZ}}{I_Z} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\frac{d}{dt}\begin{bmatrix}\beta\\ p\\ r\\ \varphi\end{bmatrix} = \begin{bmatrix} Y_\beta & \sin(\alpha) & -\cos(\alpha) & \cos(\varphi)\cos(\theta)\frac{g}{V} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & \cos(\varphi)\tan(\theta) & 0 \end{bmatrix}\begin{bmatrix}\beta\\ p\\ r\\ \varphi\end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} & Y_{\delta_1} & Y_{\delta_2} & Y_0 \\ L_{\delta_a} & L_{\delta_r} & L_{\delta_1} & L_{\delta_2} & L_0 \\ N_{\delta_a} & N_{\delta_r} & N_{\delta_1} & N_{\delta_2} & N_0 \\ 0 & 0 & 0 & 0 & \dot{\varphi}_0 \end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\\ \delta_1\\ \delta_2\\ 1\end{bmatrix} \tag{6}$$

$$a_y = \frac{V}{g}\left[\dot{\beta} - \sin(\alpha)p + \cos(\alpha)r - \cos(\varphi)\cos(\theta)\frac{g}{V}\varphi\right] + a_{y_{bias}} \tag{7}$$

The unknown parameters are contained in the matrices A, B, G, and H and in the bias vector, v. For notational simplicity, the unknown parameters will be regarded as forming a vector c. Then A, B, G, H, and v are functions of c. There is no provision for modeling state noise, that is, random or unknown inputs to the system such as turbulence. (This problem is treated in reference 2.) Instead, it is assumed that noise is introduced only in the measurement process. It is also assumed that there is no noise in the control measurements.

The integral squared error criterion can now be expressed as finding the vector of unknowns, c, that minimizes the cost functional:

$$J = \frac{1}{T} \int_0^T [z(t) - y(t)]^* D1[z(t) - y(t)]dt \tag{8}$$

or as approximated in the discrete case:

$$J = \frac{1}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D1(z_i - y_i) \tag{9}$$

where $D1$ is the symmetric, non-negative definite weighting matrix, i is a time index, and N is the number of time points. The cost functional, J, can also be called the index of performance or the fit error.

SOLUTION BY THE MODIFIED NEWTON-RAPHSON METHOD

Several algorithms for the minimization of nonlinear functionals exist that could be used to minimize J. The modified Newton-Raphson method has proved to be the most suitable for aircraft derivative determination, both in terms of computer time and convergence properties.

The Newton-Raphson algorithm is an iterative method of functional minimization which requires some initial estimate of c and a means of computing the first and second gradients of J with respect to c. Revised estimates of c are then obtained from the equation

$$c_L = c_{L-1} - \left(\nabla_c^2 J\right)_L^{-1} \left(\nabla_c J\right)_L^* \tag{10}$$

where L denotes the iteration number, ∇_c indicates the gradient with respect to c, and ∇_c^2 indicates the second gradient. The first and second gradients of J are then

$$\nabla_c J = \frac{2}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D1 \nabla_c (z_i - y_i) \tag{11}$$

$$\nabla_c^2 J = \frac{2}{N-1} \sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D1 \nabla_c (z_i - y_i) + \frac{2}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D1 \nabla_c^2 (z_i - y_i) \quad (12)$$

Computation of $\nabla_c (z_i - y_i)$ is relatively straightforward, as described in reference 3. Computation of $\nabla_c^2 (z_i - y_i)$ is much more time consuming; however, Balakrishnan shows in reference 4 that the contribution of this term to the second gradient goes to zero as the process converges. Thus, if we neglect this term, the method is still an asymptotically unbiased estimator. The Newton-Raphson algorithm with this term neglected is referred to as the modified Newton-Raphson algorithm and provides the same result as obtained by quasilinearization.

Reference 1 describes a modification in the computation of the gradient that is used on the first iteration. This modification, analogous to linear least squares, helps to obtain convergence when the initial estimates are far from the minimum. With this modification it is often possible to start with estimates of zero for all the unknowns and still converge to the correct solution.

INCLUSION OF *A PRIORI* INFORMATION

Information from wind tunnel studies, previous flight tests, and other sources (referred to collectively as predicted derivatives) is often available on the values of some of the aircraft derivatives. It may be desirable to include this information in the program's algorithm. The use of this information is particularly important when there is a linear dependence or near dependence of the effect of several derivatives, for instance, in a maneuver in which the control motion is due largely or solely to a feedback of the states. The second gradient matrix then becomes ill-conditioned, resulting in poor convergence properties and unreliable estimates. In most instances a true minimum of the cost functional is still approached, despite the ill conditioning. The location of this minimum may not be important, however, because the linearly dependent derivatives could be altered greatly without significantly increasing the cost. In this instance the slight improvement in the fit obtained by altering the derivatives would not seem sufficient justification for altering them from the *a priori* values.

One solution to this problem would be to add to the cost functional a quadratic penalty function for departure from the *a priori* values. The cost functional, J, would then be

$$J = \frac{1}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D1 (z_i - y_i) + (c - c_0)^* D2 (c - c_0) \quad (13)$$

where c_0 is the *a priori* estimate, and $D2$ is a symmetric, non-negative definite

weighting matrix. The algorithm with this penalty function will be referred to as the modified maximum likelihood estimator. It is important in this formulation for the elements of $D2$ to be small enough that, in general, $(c - c_0)^* D2(c - c_0)$ is significantly less than $\frac{1}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D1(z_i - y_i)$. Thus the estimates of those parameters that are well defined by the response data will not be altered.

The first and second gradients of J now become

$$\nabla_c J = \frac{2}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D1 \nabla_c (z_i - y_i) + 2(c - c_0)^* D2 \tag{14}$$

$$\nabla_c^2 J = \frac{2}{(N - 1)} \sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D1 \nabla_c (z_i - y_i) + 2 D2 \tag{15}$$

where the second term of equation (12) has been neglected.

When this feature is used, convergence is generally improved. With small enough values of $D2$, the estimates of the derivatives are not affected when the maneuver is well conditioned, but poorly conditioned maneuvers may converge and reveal some information instead of diverging.

CONFIDENCE LEVELS

One advantage of using a maximum likelihood estimator to determine aircraft stability and control derivatives is that an objective measure of the validity of the estimates is obtainable. With some other methods the main criterion of the validity of an estimate is the engineer's subjective judgment.

If the noise obeys the stated assumptions and $D1$ is, in fact, the inverse of the noise covariance matrix, the Cramèr-Rao inequality (ref. 3) gives a lower bound on the covariance matrix of the estimates as follows:

$$E\left[(c - c_0)(c - c_0)^*\right] \geq \left[\sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D1 \nabla_c (z_i - y_i)\right]^{-1} \tag{16}$$

The right side of this inequality is recognized as $\left(\nabla_c^2 J\right)^{-1}$ evaluated without the term for *a priori*. This expression is available in the minimization algorithm (eq. (12)), so these confidence levels (sometimes referred to in the literature as

is determined from the A matrix: LONG if A(1,2) > +0.5, LATR otherwise. This element is usually +1 in a longitudinal case or sin (ALPHA) in a lateral-directional case.

Items (2) to (11) are related to the input time histories. The signals which are input from the time histories fall into three classes: observations, controls, and extra. The observations form a vector, z, seven words long; the controls form a vector, u, four words long; and the extra signals form a vector four words long of quantities not actually used in the estimation process but useful in evaluating the quality of the maneuver. For a longitudinal case,

$$z = \left[\alpha \;\; q \;\; V \;\; \theta \;\; a_n \;\; \dot{q} \;\; a_x\right]^* \tag{17}$$

$$u = \left[\delta_e \;\; \delta_c \;\; \delta_1 \;\; \delta_2\right]^* \tag{18}$$

$$\text{Extra} = \left[\varphi \;\; \text{Altitude} \;\; \text{Mach number} \;\; \bar{q}\right]^* \tag{19}$$

and for a lateral-directional case,

$$z = \left[\beta \;\; p \;\; r \;\; \varphi \;\; a_y \;\; \dot{p} \;\; \dot{r}\right]^* \tag{20}$$

$$u = \left[\delta_a \;\; \delta_r \;\; \delta_1 \;\; \delta_2\right]^* \tag{21}$$

$$\text{Extra} = \left[\alpha \;\; V \;\; \text{Mach number} \;\; \bar{q}\right]^* \tag{22}$$

(2) CARD, TAPE-(logical) — input source for time histories. Set either CARD = T or TAPE = T. Only one of the two variables can be set to true in the NAMELIST. Default condition is TAPE = T.

(3) SPS — sample rate of input time histories (samples per second). If SPS is not set, a default value is computed from the times shown on the time histories. The times of the first two data points are subtracted and the difference rounded to the nearest 5 milliseconds. The reciprocal of this value is then used as the default value for SPS.

(4) THIN-(integer) — thinning factor for input data. If THIN = 1, every point on the file is used; if THIN = 2, every second point is used, and so forth. SPS is the sampling rate of the data before this thinning. Default value of 1 is used.

(5) NCASE — number of disjoint maneuvers to be used in obtaining one set of estimates. If two or more maneuvers were performed at approximately the same flight condition, they may be processed together to obtain a single set of estimates. Each interval will be weighted by the number of time points in the interval. Default value of 1 is used.

(6) SCALE-(seven-word vector) — scale factor for observations. The observations are multiplied by corresponding elements of SCALE when read in to compensate for any scaling errors or sign changes. Default sets all elements of the vector to 1.0.

weighting matrix. The algorithm with this penalty function will be referred to as the modified maximum likelihood estimator. It is important in this formulation for the elements of $D2$ to be small enough that, in general, $(c - c_0)^* D2(c - c_0)$ is significantly less than $\frac{1}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D1(z_i - y_i)$. Thus the estimates of those parameters that are well defined by the response data will not be altered.

The first and second gradients of J now become

$$\nabla_c J = \frac{2}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D1 \nabla_c (z_i - y_i) + 2(c - c_0)^* D2 \tag{14}$$

$$\nabla_c^2 J = \frac{2}{(N-1)} \sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D1 \nabla_c (z_i - y_i) + 2D2 \tag{15}$$

where the second term of equation (12) has been neglected.

When this feature is used, convergence is generally improved. With small enough values of $D2$, the estimates of the derivatives are not affected when the maneuver is well conditioned, but poorly conditioned maneuvers may converge and reveal some information instead of diverging.

CONFIDENCE LEVELS

One advantage of using a maximum likelihood estimator to determine aircraft stability and control derivatives is that an objective measure of the validity of the estimates is obtainable. With some other methods the main criterion of the validity of an estimate is the engineer's subjective judgment.

If the noise obeys the stated assumptions and $D1$ is, in fact, the inverse of the noise covariance matrix, the Cramèr-Rao inequality (ref. 3) gives a lower bound on the covariance matrix of the estimates as follows:

$$E\left[(c - c_0)(c - c_0)^*\right] \geq \left[\sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D1 \nabla_c (z_i - y_i)\right]^{-1} \tag{16}$$

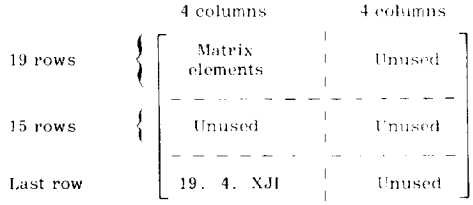
The right side of this inequality is recognized as $\left(\nabla_c^2 J\right)^{-1}$ evaluated without the term for *a priori*. This expression is available in the minimization algorithm (eq. (12)), so these confidence levels (sometimes referred to in the literature as

uncertainty levels) may be obtained with little additional effort. They can be useful in assessing the validity of the estimates obtained even when the noise characteristics are different from those assumed.

DESCRIPTION AND USE OF PROGRAMS

A basic computer program and two associated programs form a package that has been used at the NASA Flight Research Center to successfully analyze 1500 maneuvers from 20 aircraft. The basic program, referred to as the modified maximum likelihood estimation program, or MMLE, is designed to obtain maximum likelihood estimates from flight data. The associated programs, SETUP and SUMARY, although not directly related to the mathematical aspects of parameter estimation, have proved useful in extracting aircraft derivatives. The programs are designed to be used easily with the longitudinal and lateral equations of motion (eqs. (4) to (7)) by applying appropriate default values. For the options in the programs, the values designated as defaults are used only if no other values are specified. Each program is discussed in detail in the following sections.

In these programs a general matrix storage convention that permits flexibility and error checking is used. Each matrix is dimensioned with a fixed number of rows, MAX. The last row of the matrix, however, contains information about the matrix, instead of containing matrix elements. The first number in the last row is the number of rows of the matrix that are used; the second number is the number of columns used; and the third element is the matrix name in A format. For example, a 19 by 4 matrix called XJI could be stored in an array dimensioned 35 by 8 as:



This convention permits a variable-size matrix to be stored in an array of fixed dimension. The matrix manipulation subroutines can also check matrix compatibility by examining the last row before performing operations.

The programs use a standard matrix input format which facilitates data checking. The first card of any matrix to be input is a header card containing the name of the matrix, left-justified, in columns 1 to 4, the number of rows in the matrix, right-justified, in columns 9 to 10, and the number of columns in the matrix, right-justified, in columns 11 to 20. The body of the matrix follows, one row to a card, in an 8F10 format.

Additionally, the abbreviation T is used to denote true and F to denote false. NAMELIST variables follow the FORTRAN convention for type (names beginning with I, J, K, L, M, or N indicate integer variables; all other names indicate real variables), unless stated otherwise. Exceptions to this convention are given in parentheses after the NAMELIST variable.

The MMLE program can be run on most large modern computers with FORTRAN IV compilers. Approximately $31,000_{10}$ words of core storage are required. If overlay or segmentation is used, this requirement can be reduced to about $22,000_{10}$. Overlay and segmentation, however, are machine specific; directives for segmenting the MMLE program on the CDC OPERATING SYSTEM SCOPE 3.4 (ref. 5) are included in appendix A (p. 92) and can be used as a guide for implementation on other systems. Some form of automatic plotting equipment is desirable. The MMLE program plotting routines are written for a CalComp pen plotter (ref. 6). If other plotting equipment is used, it may be necessary to modify the plotting routines. The user must verify whether the routines supplied are compatible with the system being used.

From 4000 to 20,000 words of temporary disk storage are required, depending on the number of data points. This requirement is doubled if plots are made. A tape drive (two if plots are desired) may be substituted for disk storage.

Two types of input data are required for the MMLE program. The measured values contained in time histories of a flight maneuver must be available on cards, tape, or a disk file. These time histories are limited by dimensions in the plotting routines to 1000 time points per maneuver; these dimensions may be changed easily. In addition, the program must be provided information on the flight condition of the maneuver, values of pertinent characteristics of the aircraft, a set of starting estimates of the derivatives, and instructions controlling the activation of different program options.

Listings of the MMLE program and its subroutines are given in appendix A. A sample case is presented in appendix B.

Input Description

The inputs required for the MMLE program are described in this section. Each program option is explained immediately after the description of the input that controls the option.

Title card.– The title card contains any information needed to identify a particular set of data that is appropriate to include in the printed and plotted MMLE output. All 80 columns on this card may be used.

NAMELIST/INPUT/.– (See appropriate FORTRAN reference manuals for the format for specific machines.) The parameters included in the NAMELIST are as follows:

(1) LONG, LATR-(logical) – type of aerodynamic mode to be analyzed. The mode type is indicated by LONG = T or LATR = T for longitudinal or lateral-directional, respectively. Only one type should be set. If neither is set, the type

is determined from the A matrix: LONG if A(1,2) > +0.5, LATR otherwise. This element is usually +1 in a longitudinal case or sin (ALPHA) in a lateral-directional case.

Items (2) to (11) are related to the input time histories. The signals which are input from the time histories fall into three classes: observations, controls, and extra. The observations form a vector, z, seven words long; the controls form a vector, u, four words long; and the extra signals form a vector four words long of quantities not actually used in the estimation process but useful in evaluating the quality of the maneuver. For a longitudinal case,

$$z = \left[\alpha \;\; q \;\; V \;\; \theta \;\; a_n \;\; \dot{q} \;\; a_x\right]^* \tag{17}$$

$$u = \left[\delta_e \;\; \delta_c \;\; \delta_1 \;\; \delta_2\right]^* \tag{18}$$

$$\text{Extra} = \left[\varphi \;\; \text{Altitude} \;\; \text{Mach number} \;\; \bar{q}\right]^* \tag{19}$$

and for a lateral-directional case,

$$z = \left[\beta \;\; p \;\; r \;\; \varphi \;\; a_y \;\; \dot{p} \;\; \dot{r}\right]^* \tag{20}$$

$$u = \left[\delta_a \;\; \delta_r \;\; \delta_1 \;\; \delta_2\right]^* \tag{21}$$

$$\text{Extra} = \left[\alpha \;\; V \;\; \text{Mach number} \;\; \bar{q}\right]^* \tag{22}$$

(2) CARD, TAPE-(logical) — input source for time histories. Set either CARD = T or TAPE = T. Only one of the two variables can be set to true in the NAMELIST. Default condition is TAPE = T.

(3) SPS — sample rate of input time histories (samples per second). If SPS is not set, a default value is computed from the times shown on the time histories. The times of the first two data points are subtracted and the difference rounded to the nearest 5 milliseconds. The reciprocal of this value is then used as the default value for SPS.

(4) THIN-(integer) — thinning factor for input data. If THIN = 1, every point on the file is used; if THIN = 2, every second point is used, and so forth. SPS is the sampling rate of the data before this thinning. Default value of 1 is used.

(5) NCASE — number of disjoint maneuvers to be used in obtaining one set of estimates. If two or more maneuvers were performed at approximately the same flight condition, they may be processed together to obtain a single set of estimates. Each interval will be weighted by the number of time points in the interval. Default value of 1 is used.

(6) SCALE-(seven-word vector) — scale factor for observations. The observations are multiplied by corresponding elements of SCALE when read in to compensate for any scaling errors or sign changes. Default sets all elements of the vector to 1.0.

(7) FIXED-(seven-word vector) — fixed biases for observations. The known biases are added to the corresponding observations after scaling (item (6)) but before any other operations with the data. Default sets all elements of the vector to 0.

(8) DC-(four-word vector) — known biases for controls. These biases are added to the corresponding controls before any operations with the controls. Default sets all elements of the vector to 0.

(9) NREC — number of data words in each record on input tape. This parameter has no meaning if card input is used. The total number of words in each record should be at least NREC + 4, because the first four words in the record contain the time (hours, minutes, seconds, milliseconds) and are not counted as data words. (See data file input section, p. 24.) NREC is limited by program dimensions to ≤100. Default value of 15 is used unless BOTH = T (item (11)); then the value of 25 is used instead.

(10) ORDER-(15-word integer vector) — location of desired signals on input tape. This parameter has no meaning if card input is used. The signals z, u, and extra are considered to form a single vector of signals, and ORDER describes a mapping of the data record from the tape onto this vector. The I^{th} word in the resulting vector is set equal to the ORDER (I) data word in the tape record. (The first four words in the tape record contain the time and are not counted as data words.) The default is ORDER (I) = I for I = 1, 2, . . .15, which implies that there is no reordering from the input tape to the program.

(11) BOTH-(logical) — special signal order with both longitudinal and lateral-directional data on the tape. This parameter has no meaning if card input is used. If BOTH = T, the input tape is assumed to contain all the data, both longitudinal and lateral-directional, in a specific order. This order is α, q, V, θ, a_n, $\dot{q}$, a_x, δ_e, δ_c, δ_1, δ_2, φ, altitude, Mach number, $\bar{q}$, β, p, r, a_y, $\dot{p}$, $\dot{r}$, δ_a, δ_r, $\delta_{1_{lateral}}$, and $\delta_{2_{lateral}}$, where normally all angular measurements are in degrees, accelerations in g units, and velocities in feet per second. Also, if BOTH = T, NREC is overridden and set to 25; if the case is lateral-directional, the ORDER array is automatically set to [16 17 18 12 19 20 21 22 23 24 25 1 3 14 15], which overrides any order that may have been read in. Thus if the tape has data in the proper order, BOTH may be set to T and the program will automatically pick off the appropriate signals for the type of case being analyzed. Default condition is F.

Items (12) to (18) specify the form of the plotted output.

(12) PLOTEM-(logical) — time history plots comparing measured and estimated response produced if PLOTEM = T. If PLOTEM = F, no plots are made. If the *a priori* variation option (item (53)) is activated, the related derivative plots will be made instead. Default condition is T.

(13) PLTMAX — maximum error for plotting. If the error sum, J, of the last or next to last iteration is greater than PLTMAX, time history plots are not made, even

if PLOTEM = T, to avoid exceeding reasonable plotter limits. Instead, the measured time histories are printed to provide hints about the presumed problem. PLTMAX may not be larger than ERRMAX (item (22)) or it will be set equal to ERRMAX by the program. Default value of $1. \times 10^5$ is used.

(14) INCH-(logical) — plots scaled for inch grid paper if INCH = T; otherwise, for centimeter grid paper. Default condition is F.

(15) ZMIN, ZMAX-(seven-word vectors) — minimum and maximum values on vertical axis for plots comparing measured and estimated observations. The axes are 4 centimeters long (2 inches if INCH = T). If corresponding elements of ZMIN and ZMAX are equal for any signal, automatic scaling will be used on that signal. Default values are all 0 (which implies that automatic scaling is used for the default, since ZMIN = ZMAX).

(16) DCMIN, DCMAX-(eight-word vectors) — minimum and maximum values on vertical axes for plots of controls and extra signals. The comments about ZMIN and ZMAX (item (15)) apply. In addition, if automatic scaling is used for a signal and there is no nonzero point on that signal, the plot of the signal will be omitted. Default values of 0 are used.

(17) NCPLOT — number of controls and extra signals for plotting. Only the first NCPLOT controls and extra signals will be plotted in addition to the observations. This option may be used to reduce plotting of data that may be extraneous for some cases. The value of NCPLOT must be between 1 and 8, inclusive. Default value of 8 is used.

(18) TIMESC — time scale for plots in seconds per centimeter (or seconds per half inch if INCH = T). Default value of 1. is used.

(19) PRINT-(logical) — time histories based on measured data and final computed time histories printed if PRINT = T. Default condition is F.

(20) TEST-(logical) — extra output printed each iteration if TEST = T to facilitate debugging. Extra output includes time histories (in radians), the transition matrix (ref. 8) and its integral, and the first and second gradients of J. Default condition is F.

(21) NOITER — number of iterations desired. NOITER = 0 is defined as a special case for which the program computes the final time histories using initial estimates of the unknown coefficients; that is, the parameter estimation step is omitted entirely. The measured time histories are always printed when NOITER = 0, regardless of the value of PRINT (item (19)). Default value of 6 is used.

(22) ERRMAX — maximum allowable error sum. If the error sum, J, at any time becomes greater than ERRMAX, this is taken as an indication that the process is not converging properly. Therefore, iteration will stop and the measured time histories will be printed to provide clues to the reason for the problem. Default value of $1. \times 10^{20}$ is used.

(23) BOUND – convergence bound. If the error sum, J, in any iteration changes by less than BOUND times the error of the previous iteration, the process is assumed to have converged and iteration is stopped. Default value of 0.001 is used.

(24) PUNCH-(logical) – punched card output of nondimensional estimates. If PUNCH = T, the final estimates of the nondimensional derivatives are punched on cards along with the confidence levels obtained from the Cramèr-Rao bound. Default condition is F.

(25) PUNCHD-(logical) – punched card output of dimensional estimates. If PUNCHD = T, the final dimensional A and B matrices are punched on cards. These cards can be used to restart the program from the final values. Default condition is F.

(26) NEAT – number of time reductions in computation of transition matrix, $e^{A\Delta t}$, and its integral. In typical aircraft uses, a direct series evaluation of $e^{A\Delta t}$ may become computationally unstable for sample rates less than about 10 samples per second. In such cases, the power series evaluation has been used to compute $e^{A\tau}$ and its integral, with $\tau = \frac{\Delta t}{2^{\mathrm{NEAT}}}$. The desired transition matrices are then obtained after recursive applications of the formulas:

$$e^{At} = \left[e^{(At)/2}\right]^2 \tag{23}$$

$$\int_0^t e^{As}ds = \left[e^{(At)/2} + I\right]\int_0^{t/2} e^{As}ds \tag{24}$$

This process provides improved computational stability without increased time or complexity. In general, NEAT should be large enough to make $\tau \leq 0.05$ second. NEAT = 0 implies direct series computation. Default value of 0 is used.

Items (27) to (48) are related to the geometry of the aircraft and the flight condition. Items (28) to (35) are required only if nondimensional derivatives are of interest. If these items are not entered, very large values of all nondimensional derivatives will be printed as a result of the default values to avoid accidental use of the meaningless nondimensional coefficients.

(27) METRIC-(logical) – unit designation for aircraft data. If METRIC = T, all units are standard SI (MKS) units (meter, kilogram, second); otherwise, U.S. Customary (EGS) units are assumed. Default condition is F. All input data units must be consistent with the system specified.

(28) GROSWT – aircraft gross weight (pounds or newtons). Default value of $1. \times 10^9$ is used.

(29) IX-(real) – moment of inertia about the X-axis. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 1. X 10^9 is used.

(30) IY-(real) – moment of inertia about the Y-axis. This parameter is not needed for lateral-directional cases (slug-ft^2 or kg-m^2). Default value of 1. X 10^9 is used.

(31) IZ-(real) – moment of inertia about the Z-axis. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 1. X 10^9 is used.

(32) IXZ-(real) – cross-product of inertia between X- and Z-axes. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 0 is used.

(33) SPAN – wing span (ft or m). Default value of 0.001 is used.

(34) CBAR – reference chord (ft or m). Default value of 0.001 is used.

(35) S – reference wing area (ft^2 or m^2). Default value of 0.001 is used.

Items (36) to (42) concern instrument locations relative to the center of gravity. Angle-of-attack and angle-of-sideslip vane readings are corrected to the center of gravity by using the angular rates. The system model includes an arbitrary accelerometer location, so that accelerations need not be corrected to the center of gravity. The longitudinal axis locations are positive for instruments forward of the center of gravity, and the normal axis locations are positive for instruments below the center of gravity. All values are in feet or meters, and a default value of 0 is used.

(36) XB – location of angle-of-sideslip vane along the longitudinal axis.

(37) XALF – location of angle-of-attack vane along the longitudinal axis.

(38) ZB – location of angle-of-sideslip vane along the normal axis.

(39) XAY – location of a_y accelerometer along the longitudinal axis.

(40) ZAY – location of a_y accelerometer along the normal axis.

(41) XAN – location of a_n accelerometer along the longitudinal axis.

(42) ZAX – location of a_x accelerometer along the normal axis.

Items (43) to (46) are not used in the estimation process, but are useful for identifying the flight condition of the maneuver. They are passed to the SUMARY program for plot identification purposes.

(43) CG – aircraft center of gravity in fraction of chord. Default value of 0.25 is used.

(44) MACH-(real) — average Mach number. If 0, this parameter will be obtained from the input time history. Default value of 0 is used.

(45) ALPHA — average angle of attack. If 999., this parameter will be obtained from the input time history. Default value of 999. is used.

(46) PARAM — any other parameter that might be used to distinguish between flight conditions. PARAM may be used as flap position or wing sweep. Default value of 0 is used.

(47) Q — average dynamic pressure. If 0, this parameter is obtained from the input time history (lb/ft^2 or N/m^2). Default value of 0 is used.

(48) V — average velocity. If 0, this parameter is obtained from the input time history (ft/sec or m/sec). Default value of 0 is used.

(49) VAR-(three-word logical vector) — option that controls variable bias. The fifth to seventh signals of the observation vector have an unknown bias that is included in the system model. (See p. 12 for the elements of the observation vector.) This bias is determined if the corresponding elements of VAR are T. The initial values of these variable biases are 0, except for the a_n bias in a longitudinal case, which starts with a value of 1. The bias on a signal that has a D1 weighting of 0 cannot be determined; therefore, any attempt to determine a bias for an unweighted signal will be overridden in the program. Default sets all elements of the vector to T.

(50) ZERO-(four-word logical vector) — option that requires the program to determine variable initial condition. For each element of ZERO that is T, a variable increment to the initial condition is determined for the corresponding state. This increment is added to the measured initial condition to obtain the initial condition used for the computed data. If the variable initial condition is used in conjunction with NCASE > 1 (item (5)), the same increment from the measured value is used for each maneuver in the case. Default sets all elements of the vector to F.

(51) ND1, D1RLX, D1TOL — parameters that affect diagonal D1 determination option. This puts the program into a different mode of operation. A D1 weighting matrix (see matrix input section) should be determined for each airplane at the beginning of its flight program. This option automatically determines the diagonal elements of the D1 matrix based on a particular case and is activated if ND1 > 0. The program executes one run with the initial D1 matrix (described on p. 23) input, or its default, and then applies a simple iterative algorithm ND1 times to determine the proper D1 matrix. Each iteration of this algorithm involves another run through the estimation loop to obtain a set of weighted relative errors ($E_k = \frac{D1_{kk}}{t} \int_0^t \left[z_k(t) - y_k(t)\right]^2 dt$). The algorithm is designed to find a D1 matrix that results in the weighted error being approximately 1 on each signal being used (as indicated by a nonzero initial estimate of the corresponding D1 element). The motivation for this procedure is discussed in reference 3. The revised estimate of each diagonal element of the D1 matrix is then produced by multiplying the previous estimate by a factor that depends on the previous weighted error of that signal, E_k, and a relaxation factor,

D1RLX. If $E_k \geq 1$, the factor is $\frac{1}{(E_k - 1)\text{D1RLX} + 1}$; and if $E_k < 1$, the factor is $\left(\frac{1}{E_k} - 1\right)\text{D1RLX} + 1$. The variable D1TOL will stop this process if the process has converged before ND1 iterations. If, after any iteration, none of the weighted errors are greater than D1TOL or less than $\frac{1}{\text{D1TOL}}$, a final iteration will be run, and the process will be stopped. The parameter WMAPR (item (52)) will be set to 0 if this option is used, regardless of the MMLE program's input value. If plotting was specified (item (12)), only the time history using the final D1 vector will be plotted. If both the D1 vector determination and the *a priori* variation (item (53)) are activated, the D1 vector will be determined first, and the *a priori* variation will use the final D1 matrix. Default values used are ND1 = 0, D1RLX = 1.2, and D1TOL = 1.4.

Items (52) and (53) are related to the *a priori* feature.

(52) WMAPR – overall weighting factor for *a priori* information. Each element in the *a priori* weighting matrices APRA and APRB (see matrix input section) is multiplied by WMAPR before use. A value of 0 implies that the *a priori* feature is not used in the estimation process. Default value of 0 is used.

(53) NAPR, WFAC – parameters that control *a priori* variation option which puts the program into a different mode. If the *a priori* feature is used, a set of *a priori* weighting matrices should be selected at the beginning of the flight program for each aircraft analyzed. In determining the best weighting matrices to use, it is useful to run the same case with several values of WMAPR (item (52)). Reference 3 describes this process. The option to vary the value of WMAPR is activated if NAPR is greater than 0. The program then runs the entire case a total of NAPR times with different values of WMAPR. The first run is with WMAPR = 0, and the second run is with the value specified for WMAPR by item (52) (if 0 was specified, 0.001 is used instead). For each subsequent run, the value of WMAPR used is WFAC times the value used on the previous run. Time history plots are never produced when this option is used; instead, if PLOTEM = T (item (12)), the final estimates of each of the derivatives are plotted versus WMAPR on a logarithmic scale. The *a priori* estimates, which may be considered as the estimates obtained as WMAPR approaches infinity, are also plotted to the right of the other estimates. These plots may then be used as described in references 3 and 8 to estimate the best values to use for the *a priori* weightings. For these plots to be correct, the NAMELIST variable PUNCH (item (24)) must equal F, because of the order in which the computations are performed. Default values of WFAC = 100. and NAPR = 0 are used.

Time cards.– For each of the NCASE (NAMELIST item (5)) time segments to be included, one time card is required. The time cards contain the start and end times for the segment expressed as hours, minutes, seconds, and milliseconds in the format (2(3I2,I3,1X)). The program starts the segment at the first time point greater than or equal to the start time and stops it at the last point less than or equal to the stop time.

Matrix input.– Several input matrices are read next in a standard matrix input format. The matrices may be read in any order. Only the A and B matrices must be read in; the others may be read in if the default values are to be changed.

A matrix (4 by 4): The A matrix is the starting estimate of the stability matrix. For a longitudinal three-degree-of-freedom case it should be:

$$\begin{bmatrix} Z_\alpha & 1. & Z_V & -\sin(\theta)\cos(\varphi)\frac{g}{V} \\ M_\alpha & M_q & M_V & 0. \\ X_\alpha & 0. & X_V & -\cos(\theta)g \\ 0. & \cos(\varphi) & 0. & 0. \end{bmatrix}$$

In a two-degree-of-freedom case the third column should be set to 0. For a lateral-directional case the A matrix should be:

$$\begin{bmatrix} Y_\beta & \sin(\alpha) & -\cos(\alpha) & \cos(\varphi)\cos(\theta)\frac{g}{V} \\ L_\beta & L_p & L_r & 0. \\ N_\beta & N_p & N_r & 0. \\ 0. & 1. & \cos(\varphi)\tan(\theta) & 0. \end{bmatrix}$$

Average values of α, θ, φ, and V are used in these matrices.

B matrix (4 by 5 to 4 by 8): The B matrix is the starting estimate of the control matrix. The first four columns are for the control derivatives; the fifth column contains aerodynamic biases (treated as control derivatives, in which the control is defined as a constant value of 1 radian). Usually, only these five columns are required. If NCASE is greater than 1, independent aerodynamic biases may be determined for up to the first four maneuvers when necessitated by trim changes or other factors. In this event, the fifth column's aerodynamic biases are included in every maneuver, the sixth column's biases are included in all maneuvers after the first, the seventh column's biases are included in all maneuvers after the second, and the eighth column's biases are included in all maneuvers after the third. Thus the total aerodynamic bias on the first maneuver would be in column 5; for the bias on the second maneuver, columns 5 and 6 would be added; for the third maneuver, columns 5, 6, and 7 would be added; and for the fourth and all subsequent maneuvers, columns 5, 6, 7, and 8 would be added. For a lateral-directional case the B matrix should then be:

$$\begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} & Y_{\delta_1} & Y_{\delta_2} & Y_0 & Y_{0_2} & Y_{0_3} & Y_{0_4} \\ L_{\delta_a} & L_{\delta_r} & L_{\delta_1} & L_{\delta_2} & L_0 & L_{0_2} & L_{0_3} & L_{0_4} \\ N_{\delta_a} & N_{\delta_r} & N_{\delta_1} & N_{\delta_2} & N_0 & N_{0_2} & N_{0_3} & N_{0_4} \\ 0. & 0. & 0. & 0. & \dot{\varphi}_0 & \dot{\varphi}_{0_2} & \dot{\varphi}_{0_3} & \dot{\varphi}_{0_4} \end{bmatrix}$$

For a longitudinal case the B matrix would be:

$$\begin{bmatrix} Z_{\delta_e} & Z_{\delta_c} & Z_{\delta_1} & Z_{\delta_2} & Z_0 & Z_{0_2} & Z_{0_3} & Z_{0_4} \\ M_{\delta_e} & M_{\delta_c} & M_{\delta_1} & M_{\delta_2} & M_0 & M_{0_2} & M_{0_3} & M_{0_4} \\ X_{\delta_e} & X_{\delta_c} & X_{\delta_1} & X_{\delta_2} & X_0 & X_{0_2} & X_{0_3} & X_{0_4} \\ 0. & 0. & 0. & 0. & \dot{\theta}_0 & \dot{\theta}_{0_2} & \dot{\theta}_{0_3} & \dot{\theta}_{0_4} \end{bmatrix}$$

AA array (4 by 4): The AA array defines which elements in the A matrix are to be determined by the program. Each element in the AA array should be either 1. or 0.. A 1. implies that the corresponding element in the A matrix will be estimated by the program, whereas a 0. implies that it will be held fixed at the starting value. If not read in, the AA array has the following default:

Longitudinal –

$$\begin{bmatrix} 1. & 0. & 0. & 0. \\ 1. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} 1. & 1. & 0. & 0. \\ 1. & 1. & 1. & 0. \\ 1. & 1. & 1. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

BB array (4 by 5 to 4 by 8): The BB array defines which elements in the B matrix are to be determined in the same manner as the AA array defines those in the A matrix. If not read in, the BB array defaults to:

Longitudinal –

$$\begin{bmatrix} 1. & 0. & 0. & 0. & 1. \\ 1. & 0. & 0. & 0. & 1. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} 1. & 1. & 0. & 0. & 1. \\ 1. & 1. & 0. & 0. & 1. \\ 1. & 1. & 0. & 0. & 1. \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix}$$

AR matrix (4 by 4): The AR matrix is the *a priori* stability matrix and contains the *a priori* value of the A matrix. If the *a priori* feature is used, the estimates are weighted toward the AR matrix values. In general, the *a priori* values and the starting values are the same, but it is possible to distinguish between them. If not read in, the AR matrix is set equal to the A matrix.

BR matrix (4 by 5 to 4 by 8): The BR matrix is the *a priori* control matrix and plays a role similar to that of the AR matrix. If not read in, it is set equal to the B matrix.

APRA matrix (4 by 4): The APRA matrix contains *a priori* weightings for the stability matrix and contains the weightings to be assigned to the elements of the AR matrix for the *a priori* option. The program multiplies each relevant element in the APRA matrix by the overall weighting factor, WMAPR (NAMELIST item (52)), and assigns it an appropriate diagonal location in the D2 matrix (eq. (13)). No provision is made for the input of off-diagonal elements of the D2 matrix, although they are provided for in the algorithm. If not read in, the APRA matrix defaults to:

Longitudinal –

$$\begin{bmatrix} 13000. & 0. & 0. & 0. \\ 15. & 800. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} 13000. & 13000. & 13000. & 0. \\ 0.15 & 500. & 5. & 0. \\ 15. & 800. & 800. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

APRB matrix (4 by 5 to 4 by 8): The APRB matrix contains *a priori* weightings for the control matrix and plays a role analogous to that of the APRA matrix. If not read in, the APRB matrix defaults to:

Longitudinal –

$$\begin{bmatrix} 13000. & 13000. & 13000. & 13000. & 0. & 0. & 0. & 0. \\ 15. & 15. & 15. & 15. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} 13000. & 13000. & 13000. & 13000. & 0. & 0. & 0. & 0. \\ 0.15 & 0.15 & 0.15 & 0.15 & 0. & 0. & 0. & 0. \\ 15. & 15. & 15. & 15. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

AP array (3 by 4): The AP array is used in the formation of the observation matrix G of equation (2). For the aircraft identification problem, the observations generally available are either elements of the state vector, their derivatives, or accelerations. If only states and their derivatives are available, the G matrix would be identical to the A matrix. When accelerations are also of interest, the G matrix may still be expressed as a simple function of the A matrix; this function

is specified by the AP array. For example, consider the two-degree-of-freedom longitudinal case:

$$\dot{\alpha} = Z_{\alpha}\alpha + q - \frac{g}{V}\cos(\varphi)\sin(\theta)\theta + Z_{\delta_e}\delta_e + \left(Z_0 + \frac{g}{V}\cos\varphi\cos\theta\right) \qquad (25)$$

$$\left(a_n - a_{n_{bias}}\right) = -\frac{V}{g}Z_{\alpha}\alpha + 0q - 0\theta + \left(-\frac{V}{g}\right)Z_{\delta_e}\delta_e - \frac{V}{g}\left[Z_0 + \frac{g}{V}\cos\varphi\cos(\theta)\right] \qquad (26)$$

where

$$a_{n_{bias}} = -\cos\varphi\cos\theta + a_{n_{instrument\ bias}}$$

From this example it can be seen that $\left(a_n - a_{n_{bias}}\right)$ can be computed like $\dot{\alpha}$ if appropriate terms are simply multiplied by constant values of $-\frac{V}{g}$ or 0. Thus each element in the G matrix can be defined as the product of the corresponding element in $R^{-1}A$ and a constant. These constants form the AP array. This formulation results in a considerable saving of computer time. It should be noted that the accelerometer offsets from the center of gravity (NAMELIST items (39) to (42)) add terms to the G matrix after the basic terms are computed from the AP array. If the AP array is read in, the BP array must also be read in. If not read in, the AP array defaults to the following standard forms:

Longitudinal –

$$\begin{bmatrix} -\frac{V}{g} & 0. & 0. & 0. \\ 1. & 1. & 1. & 1. \\ \frac{1}{g} & 0. & 0. & 0. \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} \frac{V}{g} & 0. & 0. & 0. \\ 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. \end{bmatrix}$$

BP array (3 by 5 to 3 by 8): The BP array plays a role analogous to that of the AP array. It defines the H matrix of equation (2) as a function of the B matrix. Each element in the H matrix is defined as the product of the corresponding elements in $R^{-1}B$ and the BP array. As in the G matrix, accelerometer offsets from the center of gravity may cause additional terms to be added to the basic H matrix. If either the AP or the BP array is read in, both must be read in. The BP array defaults to:

Longitudinal –

$$\begin{bmatrix} -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} \end{bmatrix}$$

Lateral-directional –

$$\begin{bmatrix} \frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \end{bmatrix}$$

R matrix (4 by 4): The R matrix is an acceleration transformation matrix. If not read in, it defaults to the unit matrix for longitudinal cases, or for lateral-directional cases to:

$$\begin{bmatrix} 1. & 0. & 0. & 0. \\ 0. & 1. & -\frac{I_{XZ}}{I_X} & 0. \\ 0. & -\frac{I_{XZ}}{I_Z} & 1. & 0. \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

D1 matrix (5 by 5 to 7 by 7): The D1 matrix is the signal weighting matrix. The diagonal elements are the weightings used for each response signal in the cost functional. The size of this matrix determines the number of signals used in the analysis; therefore, if $\dot{p}$ and $\dot{r}$ are not measured for a lateral-directional case, the D1 matrix should be 5 by 5. This reduction will save a significant amount of computer time. If the D1 matrix is diagonal, it should be entered as a vector containing the diagonal elements. The program will then recognize that the matrix is diagonal and take advantage of this in its computations. A vector is indicated by a header card with 0 for the number of columns. The vector is then entered on one card in an 8F10 format. If not read in, the D1 matrix is assumed to be diagonal with the following values:

Longitudinal –

$$[30000. \quad 200000. \quad 0. \quad 100000. \quad 2000.]$$

Lateral –

$$[500000. \quad 1500. \quad 1000000. \quad 30000. \quad 5000.]$$

ENDCASE.– The end of the matrix input is signaled by a card with ENDCASE starting in column 1. If no more cases follow, this card should have simply END instead of ENDCASE.

Card input.– If card input was specified, the input time histories are necessary here. For each time point there should be a record of two cards containing four time words (hours, minutes, seconds, milliseconds), seven observations, four controls, and four extra signals. The order of these quantities is given in NAMELIST item (11). The format is (3I2, I4, 7F10/8F10). Normally, the angular measurements are in degrees, the accelerations in g units, and the velocities in feet per second.

Data file input.— If tape input was specified, the time histories must be on an unformatted data file (either tape or disk). The device number of this file should be specified as 4 by the control cards. This file by default has 4 time words plus 15 data words per record, as in the card input. The length of the records on this file and the order of the parameters (except for the time words) may be changed by the use of the NREC and ORDER parameters (NAMELIST items (9) and (10)); alternately, the file may be specified to be in the special BOTH form (NAMELIST item (11)). Normally, the angular measurements are in degrees, the accelerations in g units, and the velocities in feet per second.

Output Description

Three basic forms of output are available from the MMLE program: printed, plotted (time history or derivative plots), and punched card.

Printed output.— The three levels of printed output are controlled by the parameters PRINT and TEST (NAMELIST items (19) and (20)). The basic output is always printed. If PRINT = T, measured and final computed time histories are also printed. If, in addition, TEST = T, time histories in radians, the transition matrices, and the first and second gradients of J are printed every iteration. The TEST parameter is generally used only for program debugging.

Appendix B presents a listing for a sample case with only the basic output. The first page (p. 112) of the output listing summarizes the input options chosen, and the second page lists the matrices read in. The dimensional and nondimensional starting values are then summarized. An asterisk indicates the values held fixed; the other values are to be determined as unknowns in the program. Each iteration includes a printout of the revised A and B matrices, the integral squared error on each input signal, the weighted errors on each signal, and the total error sum. This iterative loop may terminate in three ways. If the error sum exceeds ERRMAX (NAMELIST item (22)) at any time, the iteration will stop immediately and the input time history will be printed (not included in appendix B). If the maximum number of iterations is reached or the process converges within the range defined by BOUND (NAMELIST item (23)), normal termination will occur. The message "ITERATION TERMINATED, ERROR WITHIN .00100 BOUND" indicates that the convergence bound caused termination in the sample case.

Confidence levels in dimensional and nondimensional form are listed next. These confidence levels are analogous to the standard deviation. Their magnitude indicates the relative confidence to be placed in estimates of the same coefficient from different maneuvers. A small confidence level for a particular derivative estimate indicates that the estimate of the derivative should be very good. Confidence levels are useful in fairing estimated derivative values.

The final page (p. 117) of the first case is a summary of the converged values. The final dimensional and nondimensional derivatives are printed in the same format as the starting values, followed by the final A and B matrices. The final integral squared errors, weighted errors, and total error sum are printed, followed by a summary of the convergence of the error sum.

If either the D1 determination option (NAMELIST item (51)) or the *a priori* variation option (NAMELIST item (53)) is activated, the program prints an appropriate message at this point and begins its second pass through the estimation loop. The output resumes from the top of the third page. This output pattern would be repeated as many times as specified by the option. If more cases follow, the same output pattern is repeated for each case.

Plotted output.— If plotting is invoked (NAMELIST item (12)), time history plots like those in appendix B will be produced. On the observation signals, the solid lines represent the flight data and the dotted lines are the computed fits. When plotting is invoked and the *a priori* variation option (NAMELIST item (53)) is being used, time history plots are not produced, but, instead, the derivative plots discussed under that option (not included in appendix B).

Punched card output.— If PUNCH = T (NAMELIST item (24)), the nondimensional A and B matrices and confidence levels are punched on cards. These cards are preceded by a header card which contains the characters LATR or LONG followed by the first 35 characters of the title card and the values of MACH, ALPHA, PARAM, and CG. These cards are in the exact format required for the SUMARY plotting program. If the case is longitudinal, a computed $\delta_{e_{trim}}$ appears in the matrix location for C_{m_0}, and C_Z appears in the location for C_{Z_0}. These quantities are of more interest in this form, although the confidence levels are not readily available. (The confidence levels punched are those for the original C_{m_0} and C_{Z_0}.) The equations used to compute these parameters are:

$$\delta_{e_{trim}} = \frac{\left(C_m + C_{m_0}\right)}{C_{m_{\delta_e}}} \tag{27}$$

$$C_Z = C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_{\delta_e}}\delta_{e_{trim}} - \cos(\theta)\cos(\varphi)\frac{W}{\bar{q}S} \tag{28}$$

These equations are valid only for a two-degree-of-freedom case with no lateral-directional cross-coupling terms.

The final dimensional A and B matrices may be output on punched cards if PUNCHD = T (NAMELIST item (25)) is specified. These matrices may be used if it is desired to restart a case from the final values and run additional iterations. If the *a priori* feature is used in the restart, the original A and B matrices should be relabeled AR and BR and inserted (see discussion of AR and BR matrices, pp. 20-21) because the *a priori* values would be different from the new starting values. Any variable bias from the original run should also be subtracted from the data using FIXED (NAMELIST item (7)) in order to start at the same values as the final iteration of the previous run.

SETUP – PREPROCESSING PROGRAM

One of the most time-consuming portions of the analysis of aircraft stability and control derivatives is the preparation of input data for the derivative estimation program. The preprocessing program, SETUP, automates much of this work and is a key element in the routine processing of a large number of cases. It can produce, at the user's option, the data file and the punched input deck for the MMLE program. Listings of the program and its subroutines are presented in appendix C. A sample case is included in appendix D.

The SETUP program reads a set of predicted derivatives to be interpolated and dimensionalized for the given flight condition. The flight condition may be specified by the user, or if appropriate data were recorded on a flight tape, the program can obtain the flight condition automatically, given only the start and stop times for the case.

When the program is used in the most automated manner, the only inputs required for each case are the start and stop times, the type of case (longitudinal or lateral-directional), and an indication of which controls were used for the particular maneuver if more than one control is relevant. Using the program in this manner requires some preparation, but only at the beginning of the flight program rather than for each case. This distinction is important when several hundred cases are being analyzed, as has been done on several aircraft.

In preparation for the most automated use of the SETUP program, the user must write four small FORTRAN subroutines. Subroutine TAPEIN reads a flight data tape, finds the time interval requested, and places the data and times from the data into two arrays. The sample included in appendix D reads an unformatted tape with time in the first four words. Subroutine RDSET provides any initialization needed for TAPEIN; in the sample case it reads the number of channels on the input tape and the channel numbers of the data needed. Subroutine COND obtains the flight condition if it is to be computed automatically instead of read in manually. The averages of each of the data channels read in are available for use in this subroutine, and the subroutine can compute the required parameters from these averages. The sample obtains ALPHA, THETA, PHI, DETRIM, Q, V, and MACH from the data channel averages. The subroutine will also compute Q and V from knots indicated airspeed and altitude, if preferred. Weight, inertia, and center of gravity are not computed in the example subroutine, although they may be computed in user-supplied versions. Subroutine COND1 reads in any data needed in subroutine COND, for instance, tables of inertia versus fuel weight. This subroutine, as given in appendix C, is a null subroutine.

Input Description

The input data and the case specifications are described in the following sections.

Options.– The options to be used are specified by the following cards. All the options begin in column 1. The cards may appear in any order (except for the START card, as noted). Only the first four characters of each card are checked.

WRITE TAPE – instructs the program to write a data file for the MMLE program. This option automatically invokes the READ TAPE option.

PUNCH DECK – instructs the program to punch a data deck for the MMLE program.

READ TAPE – instructs the program to read an input tape. This option might be specified if input tape data are needed to determine the flight condition for the punched data card deck. This instruction is redundant if WRITE TAPE was specified.

START – signals the end of the options and the start of processing. This card must be the last card in the options section.

Vehicle characteristics.– The input segment that starts here and ends at, but does not include, *User-supplied data* (p. 30) is required if PUNCH DECK was specified in the preceding options. If PUNCH DECK was not specified, this segment must not be included.

NAMELIST/WIND/: The following parameters may be input in NAMELIST format:

(1) NABP – number of angle-of-attack breakpoints for predicted derivatives. Default value of 1 is used.

(2) NMBP – number of Mach number breakpoints for predicted derivatives. Default value of 1 is used.

(3) NBP – number of sets of predicted derivatives. Each set is identified subsequently as either lateral-directional or longitudinal and as having a particular value of the extra identifying parameter PARAM (used if the data are to be separated by some other criterion, such as wing sweep or flap position). Thus if there is one longitudinal and one lateral-directional data set and no additional distinction is made, NBP = 2. Dimensions in the program restrict the value of NBP to less than or equal to 8. Default value of 1 is used.

(4) LATR, LONG-(eight-word logical vectors) – parameters that specify dynamic modes of the predicted derivatives. The type of each set of predicted derivatives should be specified by setting the corresponding element of either LATR or LONG to true. Only one of the variables can be set to true in the NAMELIST. Default type for each set is longitudinal.

(5) NCLA, NCLO – number of coefficients per lateral-directional and longitudinal data set, respectively.

(6) CGLA, CGLO – reference center of gravity for lateral-directional and longitudinal predicted derivatives in fraction of reference chord. Default value of 0.25 is used.

(7) MZLA, MZLO – number of signals for the MMLE program to analyze in lateral-directional and longitudinal cases (that is, the length of the D1 vector; see D1 matrix description, p. 23). The values must be between 5 and 7, inclusive. Default value of 5 is used.

(8) WMLA, WMLO – overall *a priori* weighting for lateral-directional and longitudinal cases (WMAPR in MMLE program). If WMLA or WMLO are not entered, the SETUP program will not read the appropriate APRA and APRB matrices discussed subsequently and will use a weighting of 0. If WMLA or WMLO are set to 0, the corresponding APRA and APRB matrices will be read by the SETUP program and punched with the MMLE program card deck, although the weighting on the matrices will still be 0. If WMLA or WMLO is set to a positive value, the APRA and APRB matrices will be read and punched normally. If WMLA or WMLO is set to a negative value, the APRA and APRB matrices will not be read and the absolute value of WMLA or WMLO will be passed to the MMLE program (using the MMLE program's defaults for the APRA and APRB matrices). In all these cases, the lateral-directional usage and longitudinal usage are independent.

(9) DEG, RAD-(logical) – parameters that specify degrees or radians for units of predicted derivatives by setting either DEG = T or RAD = T. Only one of the two variables can be set to true in the NAMELIST. The rotary derivatives are per radian regardless of this option. Default condition is DEG = T.

(10) METRIC-(logical) – parameter that specifies SI (MKS) units if true and U.S. Customary (EGS) units if false. All input data units must be consistent with the system specified. Default condition is F.

(11) BODY, STAB-(logical) – parameters that specify axis system of longitudinal predicted derivatives as body or stability. Only one of the two variables can be set to true in the NAMELIST. (Lateral-directional data are in the body axes system independent of this option.) Default condition is STAB = T.

(12) S – value of reference wing area (ft^2 or m^2).

(13) SPAN – value of reference wing span (ft or m).

(14) CBAR – value of reference wing chord (ft or m).

(15) SPS – samples per second for data file. If not specified, 0 is passed to the MMLE program which then, by default, determines SPS from the times on the data file.

(16) PUNCH-(logical) – option passed to the MMLE program to control its PUNCH (MMLE NAMELIST item (24)) option to punch cards with final estimates of the nondimensional derivatives and confidence levels. Default condition is F.

(17) XB, XALF, ZB, XAY, ZAY, XAN, ZAX – instrument locations relative to the center of gravity. The meaning of each of these parameters is the same as that given in items (36) to (42) of the MMLE NAMELIST except that, as used here, these values refer to the reference center of gravity for the predicted derivatives instead of the flight center of gravity. If 0 is entered, it is assumed that the signals have been corrected to the flight center of gravity, and no additional correction terms will be used. Default value of 0 is used.

Vehicle name: The vehicle name is specified by up to eight characters, starting in column 1. These eight characters will be used on the title card punched out for

the MMLE deck and will be included on the first line of the output from the SETUP program.

Lateral-directional weighting matrix: The lateral-directional D1 matrix is read in as a vector on one card in a 7F10 format. This vector is omitted if no lateral-directional predicted derivatives are read in. If every element is 0, the default in the MMLE program will be used.

Longitudinal weighting matrix: The longitudinal D1 matrix is read in as a vector. The comments for the lateral-directional D1 vector apply.

Lateral-directional APRA and APRB matrices: The APRA and APRB matrices for lateral-directional cases are in standard matrix input format. As mentioned above, these matrices are omitted if the WMLA parameter was not read in or was set to a negative value.

Longitudinal APRA and APRB matrices: The APRA and APRB matrices for longitudinal cases are in standard matrix format. The matrices are omitted if the WMLO parameter was not specified or was negative.

Predicted derivatives: NBP sets of predicted derivatives are required in the order specified in item (4) of SETUP NAMELIST/WIND/. Each set consists of data for NCLA or NCLO coefficients, depending on whether the set is lateral-directional or longitudinal. The data for each coefficient may be read as a function of Mach number and angle of attack, or as a function of Mach number only in the following forms.

The data for each coefficient begin with a header card containing the coefficient name in the first four columns and either a 1 or a 2 in column 10; a 1 indicates that the coefficient is a function of Mach number only, a 2 indicates that it is a function of Mach number and angle of attack. The only acceptable coefficient names are: lateral-directional – CYB, CLB, CNB, CLP, CNP, CLR, CNR, CYDA, CLDA, CNDA, CYDR, CLDR, CNDR, CYD1, CLD1, CND1, CYD2, CLD2, CND2; longitudinal (body axes) – CNA, CMA, CAA, CMQ, CNV, CMV, CAV, CNDE, CMDE, CADE, CNDC, CMDC, CADC, CND1, CMD1, CAD1, CND2, CMD2, CAD2, CN, CA; longitudinal (stability axes) – CLA, CMA, CDA, CMQ, CLV, CMV, CDV, CLDE, CMDE, CDDE, CLDC, CMDC, CDDC, CLD1, CMD1, CDD1, CLD2, CMD2, CDD2, CL, CD. The first two characters of each name indicate the force or moment coefficients (for lateral-directional, CY = side force, CL = rolling moment, and CN = yawing moment; for longitudinal, CL = lift force, CD = drag force, CN = normal force, CA = longitudinal force (positive direction is rearward)), and the remaining characters indicate the quantity with respect to which the derivative is taken. (A ~ angle of attack, B ~ angle of sideslip, P, Q, R ~ angular rates, V ~ velocity, DE, DC, DA, DR, D1, and D2 ~ controls.)

If the coefficient is a function of Mach number and angle of attack, the data for each Mach number appear on a separate card, with each card containing the values of the coefficient for the NABP angle-of-attack breakpoints. These cards are in an 8F10 format, and the card entries may be continued on additional cards if needed.

If the coefficient is a function of Mach number only, the values for the NMBP Mach number breakpoints appear on one card in an 8F10 format. As before, this card may be continued if needed.

Angle-of-attack breakpoints: A card containing the NABP values of the angle-of-attack breakpoints in an 8F10 format is necessary. If NABP = 1, this card may be blank.

Mach number breakpoints: A card containing the NMBP values of the Mach number breakpoints in an 8F10 format is necessary. If NMBP = 1, this card may be blank.

Arbitrary parameter breakpoints: A card containing the NBP values of PARAM to distinguish the predicted derivative data sets is necessary. If no distinction other than longitudinal and lateral-directional is used, this card may be blank. The card is in an 8F10 format.

User-supplied data for subroutine COND1.— Any input required for subroutine COND1 goes in the input data at this point. With the subroutine supplied, there is none.

Input tape data.— The input tape data section should be omitted if the READ TAPE option is not active. Any input required by subroutine RDSET is made here. The subroutine supplied requires a card with the number of data words per record of the input tape; this card is in an I5 format. This is followed by three cards containing the channel numbers of the 40 channels to be used; each of these cards is in a 16I5 format. A 0 indicates a signal not used. The first 25 signals will be put on the MMLE program tape if a tape is written. (The signals should be in the BOTH order defined by item (11) in the MMLE NAMELIST.) The last 15 of the 40 channels are reserved for use in subroutine COND, should they be needed. These last 15 channels are typically used for fuel weight, flap position, or any other quantities useful in identifying the flight condition and vehicle configuration. The SETUP program automatically averages the values of all 40 data channels over the requested time interval and passes these averages to subroutine COND through a labeled common block.

Case specification.— The case specification is repeated as many times as necessary, once for each case to be analyzed.

Time card: The start time and end time for the case in hours, minutes, seconds, and milliseconds are required. The format is 2(3I2,I3,1X).

NAMELIST/COND/: The following parameters may be read in NAMELIST/COND/:

(1) LONG, LATR-(logical) — type of case to be analyzed. Set either LONG or LATR to true. Only one of the two variables can be set to true in the NAMELIST.

(2) CASE-(integer) — case number. Default value of 0 is used.

(3) DELTA-(four-word logical vector) – option that specifies which controls were used in the maneuver. A value of T for any element of DELTA indicates that the corresponding control was used. If all four locations are F (default condition), the MMLE program default is used; this default is δ_e for longitudinal cases, δ_a and δ_r for lateral-directional cases. If DELTA is omitted in a case but has been specified in a previous case of the same type (longitudinal or lateral-directional), it will assume the values of the previous case.

(4) FLT-(integer) – flight number. This identification is needed only on the first case.

All the following items may be set in subroutine COND instead of reading them in at this point. The subroutine supplied will set ALPHA, THETA, PHI, DETRIM, Q, V, and MACH if the READ TAPE option is active.

(5) ALPHA – average angle of attack.

(6) THETA – average pitch attitude. Default value of 0 is used.

(7) PHI – average roll attitude. Default value of 0 is used.

(8) Q – average dynamic pressure.

(9) V – average velocity.

(10) MACH-(real) – average Mach number.

(11) PARAM – extra identifying parameter. If nonzero, the predicted derivative data with the same value of PARAM will be used for the derivatives. If there is only one longitudinal data set or one lateral-directional data set, or a longitudinal and a lateral-directional data set, PARAM need not be specified. Default value of 0 is used.

(12) W – aircraft weight (pounds or newtons).

(13) IX, IY, IZ-(real) – moments of inertia (slug-ft^2 or kg-m^2).

(14) IXZ-(real) – cross-product of inertia (slug-ft^2 or kg-m^2). Default value of 0 is used.

(15) CG – center of gravity in fraction of chord. Default is the predicted derivative reference value.

(16) DETRIM – trimmed value of δ_e. Default value of 0 is used.

Items (17) and (18) are simply for convenience if $\bar{q}$ and V are not readily available. The subroutine COND supplied may compute $\bar{q}$ and V from the values of indicated airspeed and altitude, using an approximation to the standard atmosphere.

(17) KIAS-(real) – knots indicated airspeed. If KIAS is nonzero, $\bar{q}$ and V will be computed. Default value of 0 is used.

(18) ALT – altitude (ft or m). Default value of 0 is used.

End card: The last card in the data deck contains a -1 in the first two columns to indicate the end of the data.

Output Description

The two primary outputs of the SETUP program are the MMLE program data tape and the punched card deck. These outputs are described in the MMLE Input Description section. A permanent disk file may be substituted for the data tape, without modifying the program. The punched card deck from SETUP will be ready to run through the MMLE program with the addition of control cards and the substitution of an END card for the last ENDCASE card at the end of the deck.

The printed output includes the predicted derivatives. For each case the data channel averages as passed to subroutine COND are printed if an input tape was read. All matrices punched in the MMLE program card deck are also printed for easy reference. A sample case is presented in appendix D.

SUMARY – PLOTTING PROGRAM

Data presentation can be a time-consuming portion of the derivative estimation process when a large number of maneuvers are involved. It is still common to laboriously plot derivatives and wind-tunnel data by hand, a procedure which can easily take longer than the entire estimation process. To efficiently utilize available manpower, graphs or data listings should be automatically produced. The SUMARY program produces plots of estimated derivatives and confidence levels as a function of angle of attack and, if desired, provides predicted derivative values for comparison. The program is presented as a prepared package that may be modified to meet users' specific data presentation requirements. Listings of the program and its subroutines are presented in appendix E. A sample case is given in appendix F.

The SUMARY program reads a set of predicted and flight-determined derivatives, and plots specific groups of the data as instructed. Several groups may appear on one plot, indicated by different symbols. The same predicted derivative card deck used for the SETUP program may be used in the SUMARY program, or predicted derivatives may be omitted. The flight-determined derivatives are punched out by the MMLE program in the exact format required for the SUMARY program.

Input Description

Title card.– The title card contains any information needed to identify a particular set of data that is appropriate to include in the printed output. All 80 columns on this card may be used.

NAMELIST/WIND/.— Parameters in NAMELIST/WIND/ are as follows:

(1) NABP — number of angle-of-attack breakpoints for predicted derivatives. Default value of 1 is used.

(2) NMBP — number of Mach number breakpoints for predicted derivatives. Default value of 1 is used.

(3) NBP — number of sets of predicted derivatives. The definition of a set of predicted derivatives is the same as that in the SETUP program. Default value of 1 is used.

(4) LONG, LATR-(eight-word logical vectors) — types of each set of predicted derivatives. The type is specified by setting corresponding element of either LONG or LATR to true. Only one of the two variables can be set to true in the NAMELIST. Default type for each set is longitudinal.

(5) NCLA, NCLO — number of coefficients in lateral-directional and longitudinal data sets, respectively. Default value of 0 is used.

(6) CGLA, CGLO — reference centers of gravity for lateral-directional and longitudinal predicted derivatives in fraction of chord. Default value of 0.25 is used.

(7) SHIFT-(logical) — parameter that corrects data for center-of-gravity location. If true, the flight C_{m_α} and C_{n_β} will be corrected to the predicted derivative reference center of gravity. Default condition is F.

(8) DEG, RAD-(logical) — options that specify degrees or radians for units of predicted derivatives. Only one of the two variables can be set to true in the NAMELIST. Rotary derivatives are per radian regardless of this option. Default units are degrees.

(9) BODY, STAB-(logical) — options that specify body or stability axes for input of predicted derivatives. If STAB = T, longitudinal predicted derivatives are converted from stability to body axes. If BODY = T, no conversion is made. Only one of the two variables can be set to true in the NAMELIST. Default condition is STAB = T.

(10) PRINT-(logical) — option that prints out predicted derivatives, if true. Default condition is F.

(11) WTPLOT-(logical) — option that plots predicted derivatives, if true. Default condition is T.

(12) CBAR, SPAN — aircraft reference chord and span, respectively. These quantities are needed only if SHIFT = T and there are lateral-directional data. Default values of CBAR = 0 and SPAN = 10^{50} are used.

(13) AMIN, AMAX — minimum and maximum for values on angle-of-attack axis. Default values of AMIN = 0 and AMAX = 12. are used.

(14) ASCALE – scale for angle-of-attack axis in degrees per centimeter. Default value of 1. is used.

(15) YLEN – length of ordinate axis in centimeters. Default value of 10. is used.

(16) XDIST – X-distance between plots in centimeters. Default value of 10. is used.

(17) CRFACT – factor by which confidence levels are multiplied before plotting. If equal to 0, no confidence levels are plotted. Default value of 1. is used.

(18) NPARAM – variable which distinguishes the two modes of data organization to be used. If NPARAM = 0, flight data points are sorted by Mach number to the nearest Mach number breakpoint. Plots are then produced with the different Mach numbers indicated by different symbols. If NPARAM > 0, Mach number is ignored and the data are sorted by the value of PARAM, the extra identifying parameter, to the nearest PARAM breakpoint. Plots are then produced with different symbols distinguishing these groups. The lowest Mach number of the predicted derivatives is plotted if more than one Mach number breakpoint is specified. Only one predicted derivative curve is plotted. In this case there should be only one set of lateral-directional and one set of longitudinal predicted derivatives; if there is more than one set, only the first will be plotted. Default value of 0 is used.

Predicted derivatives.– The NBP sets of predicted derivatives are necessary in exactly the same format required for the SETUP program, including the cards with angle of attack, Mach number, and PARAM breakpoints.

Flight data.– The flight data desired are required at this point in the form punched on cards by the MMLE program if PUNCH = T (p. 25).

(1) Header card – TYPE, TITLE, MACH, ALPHA, PARAM, CG in format A4,1X,A35,4F10. TYPE is either LONG or LATR.

(2) A, B, AC, BC matrices in nondimensional form. The AC and BC matrices contain the confidence levels. The fifth column of the B matrix in a longitudinal case should contain C_Z in the first row and $\delta_{e_{trim}}$ in the second row if they are desired for plotting. These quantities replace the logically expected, but more difficult to interpret, quantities (perturbation C_{Z_0} and C_{m_0}) from which they are derived.

Plotting instructions.– The end of the flight data and the beginning of the plotting instructions are signaled by a card with PLOT in the first four columns. Then, for each set of plots desired, the following instruction cards are needed:

(1) TYPE, PARM, TOL – TYPE is either LATR or LONG. PARM should equal one of the PARAM breakpoints of the predicted derivatives. The program will then select the corresponding set of predicted derivatives to be used. Flight data points with this same value of PARAM (±TOL) will be selected for plotting. For instance, if PARM = 35. and TOL = 2., a flight point with PARAM = 36. will be plotted, but

a flight point with PARAM = 38. will be rejected. In the special case, PARM = 0, the first set of predicted derivatives of the correct type (LATR, LONG) is used together with all the flight data. The format of this card is A4,F6,F10.

(2) Up to six cards specifying the derivatives to be plotted and the scales to use. Four plot instructions are included on a card (less may be on the last card). Each plot instruction is of the form DERIV, SMIN, SMAX; DERIV is the derivative name, and SMIN and SMAX are the minimum and maximum values for the ordinate. The valid derivative names are the same as those in the SETUP program for lateral-directional data; for longitudinal data, all the body axis derivative names except CA are valid and the additional name of DE may be used to plot $\delta_{e_{trim}}$ versus α_{trim}. If SMIN = SMAX (in particular, if left blank), automatic scaling will be used for the plot. The format of these cards is 4(A4,F6,F10).

End card.— The end of the plotting instructions is signaled by a card with END starting in column 1.

Output Description

The printed output from the SUMARY program includes the header cards for all flight points read in and a summary of the plotting instructions. The predicted derivatives are printed if PRINT is set to T. In addition, informative messages are provided if no predicted derivatives or flight data are available at a requested condition.

Plots are scaled for centimeter grid paper. Confidence levels are indicated by vertical bars. Predicted derivative data are identified by small symbols that correspond to those in the figure legend, at the beginning and end of each curve. A sample is shown in appendix F.

CONCLUDING REMARKS

A digital computer program written in FORTRAN IV has been successfully applied by relatively inexperienced personnel to aircraft linear parameter estimation problems with measurement noise but no state noise. This maximum likelihood estimation program includes an option for using *a priori* information and provides estimates of the derivatives and confidence levels. A program to automate the setup work and a program to plot the results have also been written. The three programs form a package which has been used to successfully analyze 1500 maneuvers on 20 aircraft.

Flight Research Center
National Aeronautics and Space Administration
Edwards, Calif., January 22, 1975

APPENDIX A

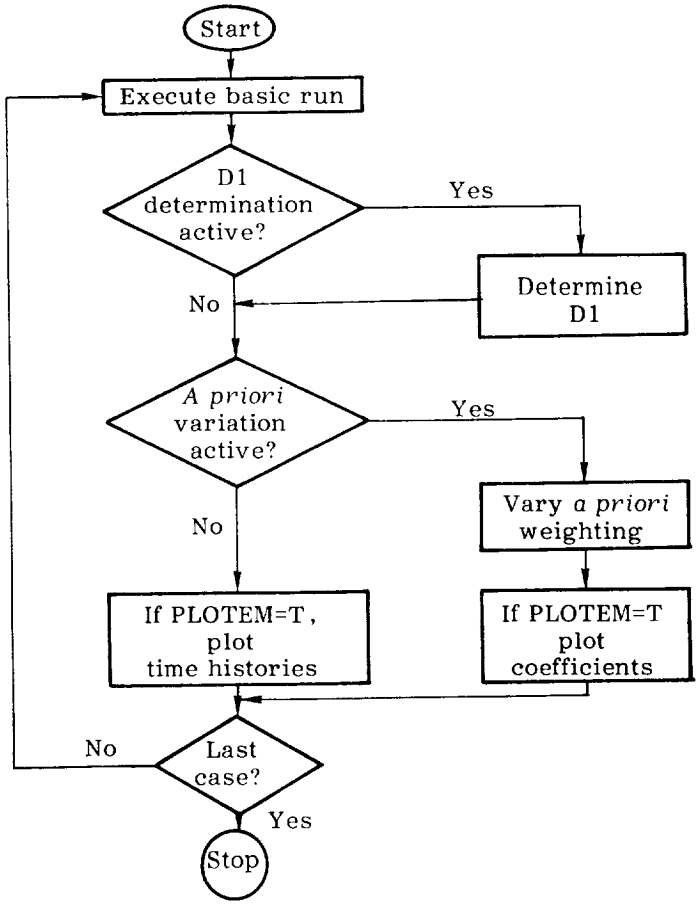
MMLE PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the MMLE program are presented. The listings are preceded by a brief description, a flow chart, when needed for clarification, and programing notes which explain some of the conventions used and point out items needed to understand the operation of the program.

MAIN MMLE PROGRAM

Description: The main MMLE program activates the three operating modes of the program (basic mode, D1 determination mode, and *a priori* variation mode).

Flow chart:



Programing notes: The PROGRAM card is required on CDC 6000/7000 systems. On an IBM 360/370 system the following DD cards, or equivalent information, are necessary to perform the same function as the PROGRAM card:

```
//GO.FT02F001 DD SYSOUT=B,SPACE=(TRK,10,RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520)
//GO.FT03F001 DD SYSOUT=A,SPACE=(TRK,50,RLSE),
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//GO.FT04F001 DD DUMMY
```

(Substitute the appropriate DD card for the input file if a tape or disk input is used.)

```
//GO.FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(10,2)),
// DCB=(RECFM=VSB,LRECL=92,BLKSIZE=924),DSN='PLOTTER DATA'
//GO.FT07F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(10,2)),
// DCB=(RECFM=VSB,LRECL=84,BLKSIZE=844),DSN='INTERNAL'
//GO.PLOTTAPE DD DUMMY
```

(Substitute the appropriate DD card for the plotter file as used on the particular system. The file name will be either PLOTTAPE or FT13F001, depending on the plotter software used.)

```
//GO.FT01F001 DD*
```

This routine alters PRINT and PLOTEM to suppress any extraneous output during intermediate steps of the D1 determination and the *a priori* variation option.

Important variables —

ND1, NAPR — control the D1 determination and *a priori* variation options as described in MMLE NAMELIST input.

D2 — vector of final weighted relative errors returned from the estimation process.

STORE — storage for final coefficient values during *a priori* variation. It is used to plot these values.

Program listing:

```
      PROGRAM MMLE(INPUT,PUNCH,OUTPUT,TAPE4,TAPE7,TAPE8,TAPE13,          MAIN   0
     -     TAPE1=INPUT,TAPE2=PUNCH,TAPE3=OUTPUT)                         MAIN  10
C                                                                        MAIN  20
C                                                                        MAIN  30
      COMMON /ALLDIM/ MAX,MIX                                            MAIN  40
      COMMON /BUF/ BUFFER,YO,THGT                                        MAIN  50
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                              MAIN  60
      COMMON /COM/ NCASE,MZ,NPTS,NPT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,     MAIN  70
     -     D1TOL,D1RLX,NAPR,WFAC,WMAPR,ERRSUM,LAST,RATIO                 MAIN  80
      DIMENSION NPTS(15),D1(8,7),AHOLD(5,4),                             MAIN  90
     -     BHOLD(5,8),D2(7),A(5,4),B(5,8),AP(4,4),BP(4,8),STORE(14,27)   MAIN 100
     -     ,BUFFER(1024),RI(5,4)                                         MAIN 110
      LOGICAL PLOTEM,LONG,LATR,PLT,AA(5,4),BB(5,8),STP,PRINT,PRNT,       MAIN 120
     -     FIRST,LAST                                                    MAIN 130
      FIRST=.TRUE.                                                       MAIN 140
      LAST=.FALSE.                                                       MAIN 150
    5 REWIND 4                                                           MAIN 160
      REWIND 8                                                           MAIN 170
      REWIND 7                                                           MAIN 180
      CALL EDIT                                                          MAIN 190
      IF(ND1.EQ.0.AND.NAPR.LT.1) GO TO 10                                MAIN 200
      PLT=PLOTEM                                                         MAIN 210
      IF(NAPR.GT.0) PLOTEM=.FALSE.                                       MAIN 220
      PRNT=PRINT                                                         MAIN 230
      WHOLD=WMAPR                                                        MAIN 240
      WMAPR=0.                                                           MAIN 250
      MAX=5                                                              MAIN 260
      CALL AMAKE(AHOLD,A)                                                MAIN 270
      CALL AMAKE(BHOLD,B)                                                MAIN 280
C************************************ BASE RUN                            MAIN 290
   10 CALL DATA(.TRUE.)                                                  MAIN 300
      CALL AGIRL                                                         MAIN 310
      IF(ND1.EQ.0) GO TO 15                                              MAIN 320
      PLOTEM=.FALSE.                                                     MAIN 330
      PRINT=.FALSE.                                                      MAIN 340
   15 CALL OUTPUT(D2)                                                    MAIN 350
      IF (ND1.EQ.0) GO TO 100                                            MAIN 360
C************************************ D1 DETERMINATION (IF DESIRED)       MAIN 370
      TOLI=1./D1TOL                                                      MAIN 380
      DO 90 I=1,ND1                                                      MAIN 390
      STP=.TRUE.                                                         MAIN 400
      DO 30 J=1,MZ                                                       MAIN 410
      IF(D1     (J,J).NE.0.) GO TO 22                                    MAIN 420
      D2(J)=1.                                                           MAIN 430
      GO TO 30                                                           MAIN 440
   22 IF(D2(J).GT.D1TOL.OR.D2(J).LT.TOLI) STP=.FALSE.                    MAIN 450
      IF(D2(J).GT.1.) GO TO 25                                           MAIN 460
      D2(J)=(1./D2(J)-1.)*D1RLX+1.                                       MAIN 470
      GO TO 27                                                           MAIN 480
   25 D2(J)=1./((D2(J)-1.)*D1RLX+1.)                                     MAIN 490
   27 D1     (J,J)=D1     (J,J)*D2(J)                                    MAIN 500
      D2(J)=SQRT(D2(J))                                                  MAIN 510
   30 CONTINUE                                                           MAIN 520
      IF(I.LT.ND1.AND..NOT.STP) GO TO 40                                 MAIN 530
      PLOTEM=PLT                                                         MAIN 540
      PRINT=PRNT                                                         MAIN 550
   40 WRITE(3,2000)                                                      MAIN 560
```

```
      MAX=8                                                              MAIN 570
      CALL ASPIT(D1)                                                     MAIN 580
      MAX=5                                                              MAIN 590
      CALL AMAKE(A,AHOLD)                                                MAIN 600
      CALL AMAKE(B,BHOLD)                                                MAIN 610
      CALL DATA(.FALSE.)                                                 MAIN 620
      CALL AGIRL                                                         MAIN 630
      CALL OUTPUT(D2)                                                    MAIN 640
      IF(STP) GO TO 95                                                   MAIN 650
   90 CONTINUE                                                           MAIN 660
   95 WRITE(3,2002)                                                      MAIN 670
      MAX=8                                                              MAIN 680
      CALL ASPIT(D1)                                                     MAIN 690
  100 IF(NAPR.GT.0) GO TO 105                                            MAIN 700
      IF(.NOT.PLOTEM) GO TO 200                                          MAIN 710
      CALL THPLOT(FIRST)                                                 MAIN 720
      FIRST=.FALSE.                                                      MAIN 730
      GO TO 200                                                          MAIN 740
C************************************ APRIORI VARIATION (IF DESIRED)      MAIN 750
  105 PRINT=.FALSE.                                                      MAIN 760
      IF(WHOLD.EQ.0.) WHOLD=.001                                         MAIN 770
      WMAPR=WHOLD                                                        MAIN 780
      I=0                                                                MAIN 790
  110 I=I+1                                                              MAIN 800
      JKMM=0                                                             MAIN 810
      DO 130 J=1,3                                                       MAIN 820
      DO 120 K=1,4                                                       MAIN 830
      IF(BB(J,K)) GO TO 120                                              MAIN 840
      JKMM=JKMM+1                                                        MAIN 850
      STORE(I,JKMM)=B(J,K)                                               MAIN 860
  120 CONTINUE                                                           MAIN 870
      DO 130 K=1,3                                                       MAIN 880
      IF(AA(J,K)) GO TO 130                                              MAIN 890
      JKMM=JKMM+1                                                        MAIN 900
      STORE(I,JKMM)=A(J,K)                                               MAIN 910
  130 CONTINUE                                                           MAIN 920
      STORE(I,JKMM+1)=ERRSUM                                             MAIN 930
      MAX=5                                                              MAIN 940
      CALL AMAKE(A,AHOLD)                                                MAIN 950
      CALL AMAKE(B,BHOLD)                                                MAIN 960
      IF(I-NAPR) 140,110,160                                             MAIN 970
  140 WRITE(3,2001)WMAPR                                                 MAIN 980
      CALL DATA(.FALSE.)                                                 MAIN 990
      CALL AGIRL                                                         MAIN1000
      CALL OUTPUT(D2)                                                    MAIN1010
      WMAPR=WMAPR*WFAC                                                   MAIN1020
      GO TO 110                                                          MAIN1030
  160 IF(.NOT.PLT) GO TO 200                                             MAIN1040
      CALL APRPLT(STORE,AA,BB,NAPR,WHOLD,WFAC,LONG,FIRST,LAST,RATIO)     MAIN1050
      FIRST=.FALSE.                                                      MAIN1060
  200 IF(.NOT.LAST) GO TO 5                                              MAIN1070
      IF(.NOT.FIRST) CALL PLOT(0.,0.,999)                                MAIN1080
 2000 FORMAT(15H0D1 REVISED TO:)                                         MAIN1090
 2001 FORMAT(12H0WMAPR NOW =,E10.2)                                      MAIN1100
 2002 FORMAT(10H0FINAL D1:)                                              MAIN1110
      STOP                                                               MAIN1120
      END                                                                MAIN1130
```

SUBROUTINE EDIT

Description: Subroutine EDIT initializes the program, sets defaults, and reads input options and matrices.

Programing notes: If used with a system that does not support the NAMELIST, some other form of input might be used.

Subroutine MATLD, called at card 1820, sets appropriate elements of ABC to -99999 when reading a matrix. These elements are then tested after all the matrix input has been made to determine what matrix defaults are needed.

The R matrix is inverted at card 2460, since R^{-1} is the form needed by the rest of the program.

From card 2530 on, the AA and BB matrices are being converted to logical variables and the number of the different types of unknown coefficients to be determined is found. An element in AA or BB is set to false if that element in A or B is to be determined. This may be contrary to the expected convention.

Subroutine listing:

```
      SUBROUTINE EDIT                                                    EDIT   0
C                                                                        EDIT  10
C     SETS DEFAULT VALUES AND READS PROGRAM OPTIONS FROM CARDS           EDIT  20
C                                                                        EDIT  30
      COMMON /ALLDIM/ MAX,MIX                                            EDIT  40
      COMMON /TOPLOT/ ZMAX,ZMIN,DCMAX,DCMIN,TIMESC,NCPLOT                EDIT  50
      COMMON /TOGIRL/JKM,JKMM,JKMM1,ERRMAX,ZEROIN,XT5,BOUND,APR,NI       EDIT  60
     -     ,VAR,ZERO,APRD,JKV,DIAG                                       EDIT  70
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,R                               EDIT  80
      COMMON /COM/ NCASE,MZ,NPTS,NPTT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,    EDIT  90
     -     D1TOL,D1RLX,NAPR,WFAC,WMAPR,ERRSUM,LAST,RATIO                 EDIT 100
      COMMON /INFO/ HH,NOITER,MX,MXP1,MU,TEST,XBN,ZBX,IPQR,IXYZ,XT3,MZM,EDIT 110
     -  CORECT,BIASKN,PLTMAX,XT4,ERRVEC,PUNCHD,NEAT                      EDIT 120
      COMMON /HEADNG/ LABELS,TITLE,JULIAN                                EDIT 130
      COMMON /TOOATA/ S,SPAN,XB,ZB,XAY,CARD,ZAY,APRA,APRB,DC,            EDIT 140
     -     XALF,THIN,TAPE,CBAR,APBP,STC,ETC,FIXED,AR,BR,XAN,ZAX,         EDIT 150
     -     SCALE,NREC,ORDER,METRIC,IX,IY,IZ,IXZ,Q,V,GROSWT               EDIT 160
      COMMON /ROUTH/ PUNCH ,PARAM,MACH,ALPHA,CG,AC,BC                    EDIT 170
      INTEGER STC(15),ETC(15),THIN,ST(4),ET(4),                          EDIT 180
     -     LABELS(15),LONLAB(15),LATLAB(15),ORDER(15)                    EDIT 190
      LOGICAL CARD,TAPE,CORECT,METRIC,APBP,PUNCH,ZEROIN,PUNCHD,          EDIT 200
     -         LAA(5,4),LBB(5,8),LONG,LATR,VAR(3),ZERO(4),LAST,          EDIT 210
     -     BIASKN,PLOTEM,TEST,PRINT,BOTH,DIAG,INCH                       EDIT 220
      REAL IX,IY,IZ,IXZ,MACH,MATRX(8,8),JULIAN,AC(5,4),BC(5,8),LAB(12)   EDIT 230
      DIMENSION A(5,4),B(5,8),TITLE(20),DCMAX(8),DCMIN(8),XT4(3),        EDIT 240
     -     AA(5,4),BB(5,8),AP(4,4),BP(4,8),NPTS(15),D1(8,7),APRD(35),    EDIT 250
     -               ZMIN(7)  ,ZMAX(7)  ,ERRVEC(20),DC(4),APR(35,35),    EDIT 260
     -     XT5(35),AR(5,4),BR(5,8),APRA(5,4),APRB(5,8),ABC(12),FIXED(7),EDIT 270
     -     APRLON(5,4),APRLAT(5,4),BPRLON(5,8),BPRLAT(5,8),TLAT(3),     EDIT 280
     -     TLON(3),TYPE(3),SCALE(7),R(5,4),XT3(4),AALAT(5,4),AALON(5,4),EDIT 290
     -     BBLAT(5,8),BBLON(5,8),D1LON(5),D1LAT(5)                       EDIT 300
      EQUIVALENCE (AA(1,1),LAA(1,1)),(BB(1,1),LBB(1,1))                  EDIT 310
      DATA LONLAB/4HALFA,1HQ,1HV,4HTHET,2HAN,4HQDOT,2HAX,2HDE,2HDC,      EDIT 320
     -     3HDC1 ,3HDC2 ,3HPHI,3HALT,4HMACH,4HQBAR/,LATLAB/4HBETA,1HP,   EDIT 330
     -     1HR,3HPHI,2HAY,4HPDOT,4HRDOT,2HDA,2HDR,3HDC1,3HDC2,4HALFA     EDIT 340
     -     ,1HV,4HMACH,4HQBAR/,LAB/1HA,1HB,2HAA,2HBB,2HAR,2HBR,4HAPRA,   EDIT 350
     -     4HAPRB,2HD1,2HAP,2HBP,1HR/,SUML/3HSUM/,TLAT/4HLATE,3HRAL,     EDIT 360
     -     1H /,TLON/4HLONG,4HITUD,4HINAL/,AALAT/3*1.,0.,4.,3*1.,0.,4.,  EDIT 370
     -     0.,1.,1.,7*0./,BBLAT/3*1.,0.,4.,3*1.,0.,8.,10*0.,4*1.,16*0./,EDIT 380
     -     AALON/1.,1.,0.,0.,4.,0.,1.,0.,0.,4.,10*0./,BBLON/1.,1.,0.,0.,EDIT 390
     -     4.,4*0.,8.,10*0.,1.,1.,0.,1.,16*0./,D1LON/30000.,200000.,0., EDIT 400
     -     100000.,2000./, APRLON/13000.,15.,2*0.,4.,0.,800.,2*0.,4.,    EDIT 410
     -     10*0./,APRLAT/13000.,.15,15.,0.,4.,13000.,500.,800.,0.,4.,    EDIT 420
     -     13000.,5.,800.,7*0./,BPRLAT/13000.,.15,15.,0.,4.,13000.,.15,  EDIT 430
     -     15.,0.,8.,13000.,.15,15.,2*0.,13000.,.15,15.,22*0./,           EDIT 440
     -     BPRLON/13000.,15.,2*0.,4.,13000.,15.,2*0.,8.,13000.,15.,3*0.,EDIT 450
     -    13000.,15.,23*0./,D1LAT/500000.,1500.,1000000.,30000.,5000./ EDIT 460
      NAMELIST /INPUT/  GROSWT,Q,S,SPAN,CBAR,V,IX,IY,IZ,IXZ,PUNCHD,     EDIT 470
     -          XB,ZB,XAY,ZAY,XALF,XAN,ZAX,WMAPR,PLTMAX,NEAT,           EDIT 480
     -     CG,MACH,ALPHA,PARAM,SPS,NCASE,NOITER,TEST,PLOTEM,TIMESC,     EDIT 490
     -     PUNCH,THIN,LONG,LATR,WFAC,ND1,CARD,TAPE,BOUND,ERRMAX,        EDIT 500
     -     METRIC,PRINT,NAPR,D1RLX,D1TOL,FIXED,VAR,DC,INCH,             EDIT 510
     -     ZERO,SCALE,ZMIN,ZMAX,DCMIN,DCMAX,NREC,ORDER,BOTH,NCPLOT      EDIT 520
      JULIAN=DATE(JULIAN)                                                EDIT 530
      APR(35,3)=SUML                                                     EDIT 540
      READ (1,2000) TITLE                                                EDIT 550
      WRITE(3,2005)TITLE,JULIAN                                          EDIT 560
```

```
C************************************ DEFAULTS                          EDIT 570
      NCPLOT=8                                                        EDIT 580
      V=0.                                                            EDIT 590
      Q=0.                                                            EDIT 600
      MACH=0.                                                         EDIT 610
      PUNCHD=.FALSE.                                                  EDIT 620
      NEAT=0                                                          EDIT 630
      NREC=15                                                         EDIT 640
      METRIC=.FALSE.                                                  EDIT 650
      BOTH=.FALSE.                                                    EDIT 660
      DO 14 I=1,15                                                    EDIT 670
   14 ORDER(I)=I                                                      EDIT 680
      DO 10 I=1,8                                                     EDIT 690
      DCMIN(I)=0.                                                     EDIT 700
   10 DCMAX(I)=0.                                                     EDIT 710
      DO 13 I=1,12                                                    EDIT 720
   13 ABC(I)=LAB(I)                                                   EDIT 730
      CORECT=.FALSE.                                                  EDIT 740
      CARD=.FALSE.                                                    EDIT 750
      TAPE=.TRUE.                                                     EDIT 760
      ZEROIN=.FALSE.                                                  EDIT 770
      BIASKN=.FALSE.                                                  EDIT 780
      PLOTEM=.TRUE.                                                   EDIT 790
      TEST=.FALSE.                                                    EDIT 800
      LONG=.FALSE.                                                    EDIT 810
      LATR=.FALSE.                                                    EDIT 820
      DIAG=.TRUE.                                                     EDIT 830
      PLTMAX=1.E+05                                                   EDIT 840
      ERRMAX=1.E+20                                                   EDIT 850
      PUNCH=.FALSE.                                                   EDIT 860
      PARAM = 0.                                                      EDIT 870
      CG = .25                                                        EDIT 880
      XB=0.                                                           EDIT 890
      ZB=0.                                                           EDIT 900
      XAY=0.                                                          EDIT 910
      ZAY=0.                                                          EDIT 920
      THIN=1                                                          EDIT 930
      D1(8,1)=5.                                                      EDIT 940
      D1(8,2)=5.                                                      EDIT 950
      MAX=8                                                           EDIT 960
      CALL AZOT(D1)                                                   EDIT 970
      MAX=5                                                           EDIT 980
      MIX=5                                                           EDIT 990
      R(5,1)=4.                                                       EDIT1000
      R(5,2)=4.                                                       EDIT1010
      R(5,3)=ABC(12)                                                  EDIT1020
      CALL AZOT(R)                                                    EDIT1030
      DO 136 I=1,4                                                    EDIT1040
      R(I,I)=1.                                                       EDIT1050
      ZERO(I)=.FALSE.                                                 EDIT1060
  136 DC(I) = 0.                                                      EDIT1070
      XALF=0.                                                         EDIT1080
      ALPHA=999.                                                      EDIT1090
      PRINT=.FALSE.                                                   EDIT1100
      DO 137 I=1,7                                                    EDIT1110
      ZMIN(I)=0.                                                      EDIT1120
      ZMAX(I)=0.                                                      EDIT1130
```

```
      FIXED(I)=0.                                                       EDIT1140
  137 SCALE(I)=1.                                                       EDIT1150
      XAN=0.                                                            EDIT1160
      ZAX=0.                                                            EDIT1170
      BOUND=.001                                                        EDIT1180
      SPS=0.                                                            EDIT1190
      WMAPR=0.                                                          EDIT1200
      NOITER=6                                                          EDIT1210
      NCASE=1                                                           EDIT1220
      NI=35                                                             EDIT1230
      TIMESC=1.                                                         EDIT1240
      DO 11 I=1,3                                                       EDIT1250
      VAR(I)=.TRUE.                                                     EDIT1260
   11 TYPE(I)=TLAT(I)                                                   EDIT1270
      S=.001                                                            EDIT1280
      SPAN=.001                                                         EDIT1290
      CBAR=.001                                                         EDIT1300
      GROSWT=1.E+09                                                     EDIT1310
      IY=1.E+09                                                         EDIT1320
      IX=1.E+09                                                         EDIT1330
      IZ=1.E+09                                                         EDIT1340
      IXZ=0.                                                            EDIT1350
      NO1=0                                                             EDIT1360
      NAPR=0                                                            EDIT1370
      WFAC=100.                                                         EDIT1380
      D1RLX=1.2                                                         EDIT1390
      D1TOL=1.4                                                         EDIT1400
      INCH=.FALSE.                                                      EDIT1410
      RATIO=.7874                                                       EDIT1420
C************************************ READ PROGRAM OPTIONS               EDIT1430
      READ (1,INPUT)                                                    EDIT1440
      IF(INCH) RATIO=1.                                                 EDIT1450
      IF(NOITER.EQ.0) PUNCH=.FALSE.                                     EDIT1460
      IF(NOITER.EQ.0) PUNCHD=.FALSE.                                    EDIT1470
      PLTMAX=AMIN1(PLTMAX,ERRMAX)                                       EDIT1480
      IF(CARD) TAPE=.FALSE.                                             EDIT1490
      ATHIN=THIN                                                        EDIT1500
      IF(SPS.NE.0.) HH=ATHIN/SPS                                        EDIT1510
      IF(BOTH) NREC=25                                                  EDIT1520
      IF (LONG) GO TO 5                                                 EDIT1530
      DO 2 I=1,15                                                       EDIT1540
    2 LABELS(I)=LATLAB(I)                                               EDIT1550
      GO TO 8                                                           EDIT1560
    5 DO 6 I=1,15                                                       EDIT1570
    6 LABELS(I)=LONLAB(I)                                               EDIT1580
      DO 12 I=1,3                                                       EDIT1590
   12 TYPE(I)=TLON(I)                                                   EDIT1600
    8 CONTINUE                                                          EDIT1610
      ZEROIN = ZERO(1).OR.ZERO(2).OR.ZERO(3).OR.ZERO(4)                 EDIT1620
      BIASKN = VAR(1).OR.VAR(2).OR.VAR(3)                               EDIT1630
      WRITE(3,2009)TYPE,CARD,TAPE,SPS,THIN,NREC,BOTH                    EDIT1640
      WRITE(3,2010)WMAPR,NEAT,NOITER,BOUND,ERRMAX                       EDIT1650
      IF(NO1.NE.0) WRITE(3,2006)NO1,C1RLX,D1TOL                         EDIT1660
      IF(NAPR.NE.0) WRITE(3,2008)NAPR,WMAPR,WFAC                        EDIT1670
      WRITE(3,2011)PLOTEM,PLTMAX,NCPLOT,TIMESC,PRINT,TEST,PUNCH,PUNCHD  EDIT1680
      WRITE(3,2012)METRIC,Q,V,MACH,ALPHA,CG,PARAM,S,SPAN,CBAR,IX,IY,IZ, EDIT1690
           IXZ,GROSWT,XALF,XAN,XB,XAY,ZB,ZAY                            EDIT1700
```

```
      WRITE(3,2013)LABELS,VAR,ZERO,FIXED,DC,SCALE                        EDIT1710
      IF(PLOTEM) WRITE(3,2001)ZMIN,DCMIN,ZMA),DCMAX                      EDIT1720
      SPS=SPS/ATHIN                                                      EDIT1730
      DO 100 I=1,NCASE                                                   EDIT1740
      READ (1,1000) ST,ET                                                EDIT1750
      STC(I)=ST(4)+100C*(ST(3)+60*ST(2)+3600*ST(1))                      EDIT1760
      ETC(I)=ET(4)+100C*(ET(3)+60*ET(2)+3600*ET(1))                      EDIT1770
  100 WRITE(3,2002)I,ST,ET                                               EDIT1780
C************************************ READ MATRICES                      EDIT1790
      WRITE(3,2003)     TITLE,JULIAN                                     EDIT1800
      WRITE(3,2004)                                                      EDIT1810
      MAX=8                                                              EDIT1820
  101 CALL MATLD(MATRX,ABC,ILD)                                          EDIT1830
      IF(IABS(ILD).EC.999) GO TO 108                                     EDIT1840
      IF(ILD.EQ.9) DIAG=.FALSE.                                          EDIT1850
      IF(ILD.EQ.1) CALL MAK(A   ,MATRX,5)                                EDIT1860
      IF(ILD.EQ.2) CALL MAK(B   ,MATRX,5)                                EDIT1870
      IF(ILD.EQ.3) CALL MAK(AA  ,MATRX,5)                                EDIT1880
      IF(ILD.EQ.4) CALL MAK(BB  ,MATRX,5)                                EDIT1890
      IF(ILD.EQ.5) CALL MAK(AR  ,MATRX,5)                                EDIT1900
      IF(ILD.EQ.6) CALL MAK(BR  ,MATRX,5)                                EDIT1910
      IF(ILD.EQ.7) CALL MAK(APRA,MATRX,5)                                EDIT1920
      IF(ILD.EQ.8) CALL MAK(APRB,MATRX,5)                                EDIT1930
      IF(IABS(ILD).EQ.9) CALL MAK(D1,MATRX,8)                            EDIT1940
      IF(ILD.EQ.10) CALL MAK(AP  ,MATRX,4)                               EDIT1950
      IF(ILD.EQ.11) CALL MAK(BP  ,MATRX,4)                               EDIT1960
      IF(ILD.EQ.12) CALL MAK(R,MATRX,5)                                  EDIT1970
      GO TO 101                                                          EDIT1980
  108 MAX=5                                                              EDIT1990
      MZ=D1(8,1)                                                         EDIT2000
      APBP=.FALSE.                                                       EDIT2010
      IF(ABC(10).EQ.-99999. .AND. ABC(11).EQ.-99999.) APBP=.TRUE.        EDIT2020
      IF(LATR.OR.LONG) GO TO 117                                         EDIT2030
      IF(A(1,2).GT. .5) GO TO 113                                        EDIT2040
      LATR=.TRUE.                                                        EDIT2050
      GO TO 117                                                          EDIT2060
  113 LONG=.TRUE.                                                        EDIT2070
      DO 114 I=1,12                                                      EDIT2080
  114 LABELS(I)=LONLAB(I)                                                EDIT2090
      WRITE(3,2007)                                                      EDIT2100
  117 IF(ILD.EQ.-999) LAST=.TRUE.                                        EDIT2110
      IF(ABC(5).NE.-99999.) CALL AMAKE(AR,A)                             EDIT2120
      IF(ABC(6).NE.-99999.) CALL AMAKE(BR,B)                             EDIT2130
      IF(LONG) GO TO 121                                                 EDIT2140
      IF(ABC(9).EQ.-99999.) GO TO 119                                    EDIT2150
      DO 118 I=1,5                                                       EDIT2160
  118 D1(I,I)=D1LAT(I)                                                   EDIT2170
  119 IF(ABC(7).NE.-99999.) CALL AMAKE(APRA,APRLAT)                      EDIT2180
      IF(ABC(8).NE.-99999.) CALL AMAKE(APRB,BPRLAT)                      EDIT2190
      IF(ABC(3).NE.-99999.) CALL AMAKE(AA,AALAT)                         EDIT2200
      IF(ABC(4).NE.-99999.) CALL AMAKE(BB,BBLAT)                         EDIT2210
      IF(XB.NE.0. .OR.ZB.NE.0. .OR.XAY.NE.0. .OR.ZAY.NE.0.)CORECT=.TRUE.EDIT2220
      IF(ABC(12).EQ.-99999.) GO TO 123                                   EDIT2230
      R(2,3)=-IXZ/IX                                                     EDIT2240
      R(3,2)=-IXZ/IZ                                                     EDIT2250
      IF(TEST) CALL ASPIT(R)                                             EDIT2260
  123 IF(.NOT.BOTH) GO TO 122                                            EDIT2270
```

```
      DO 7 I=1,3                                                        EDIT2280
    7 ORDER(I)=I+15                                                     EDIT2290
      ORDER(4)=12                                                       EDIT2300
      DO 9 I=5,11                                                       EDIT2310
    9 ORDER(I)=I+14                                                     EDIT2320
      ORDER(12)=1                                                       EDIT2330
      ORDER(13)=3                                                       EDIT2340
      GO TO 122                                                         EDIT2350
  121 IF(ABC(7).NE.-99999.) CALL AMAKE(APRA,APRLON)                     EDIT2360
      IF(ABC(8).NE.-99999.) CALL AMAKE(APRB,BPRLON)                     EDIT2370
      IF(ABC(3).NE.-99999.) CALL AMAKE(AA,AALON)                        EDIT2380
      IF(ABC(4).NE.-99999.) CALL AMAKE(BB,BBLON)                        EDIT2390
      IF(XALF.NE.0. .OR. XAN.NE.0. .OR. ZAX.NE.0.) CORECT=.TRUE.        EDIT2400
      IF(ABC(9).EQ.-99999.) GO TO 122                                   EDIT2410
      DO 124 I=1,5                                                      EDIT2420
  124 D1(I,I)=D1LON(I)                                                  EDIT2430
C************************************* COMPUTE SIZE OF SYSTEM            EDIT2440
C************************************* AA AND BB TO LOGICAL VARIABLES    EDIT2450
  122 CALL INV(R,MAX)                                                   EDIT2460
      MX=A(MAX,2)                                                       EDIT2470
      MU=B(MAX,2)                                                       EDIT2480
      MXP1=MX+1                                                         EDIT2490
      MZM=MZ-MX                                                         EDIT2500
      DO 150 I=MXP1,MZ                                                  EDIT2510
  150 IF(D1(I,I).EQ.0.) VAR(I-MX)=.FALSE.                               EDIT2520
      JKMM1=0                                                           EDIT2530
      DO 120 I=1,MX                                                     EDIT2540
      DO 110 J=1,MX                                                     EDIT2550
      IF(AA(I,J)) 107,106,107                                           EDIT2560
  106 LAA(I,J)=.TRUE.                                                   EDIT2570
      GO TO 110                                                         EDIT2580
  107 LAA(I,J)=.FALSE.                                                  EDIT2590
      JKMM1=JKMM1+1                                                     EDIT2600
  110 CONTINUE                                                          EDIT2610
      DO 120 J=1,MU                                                     EDIT2620
      IF(BB(I,J)) 112,111,112                                           EDIT2630
  111 LBB(I,J)=.TRUE.                                                   EDIT2640
      GO TO 120                                                         EDIT2650
  112 LBB(I,J)=.FALSE.                                                  EDIT2660
      JKMM1=JKMM1+1                                                     EDIT2670
  120 CONTINUE                                                          EDIT2680
      JKMM=JKMM1                                                        EDIT2690
      DO 125 I=1,MX                                                     EDIT2700
  125 IF(ZERO(I)) JKMM1=JKMM1+1                                         EDIT2710
      JKV=JKMM1                                                         EDIT2720
      DO 126 I=1,MZM                                                    EDIT2730
  126 IF(VAR(I)) JKMM1=JKMM1+1                                          EDIT2740
      JKM=JKMM1+1                                                       EDIT2750
      APR(35,1)=JKMM1                                                   EDIT2760
      APR(35,2)=JKM                                                     EDIT2770
      RETURN                                                            EDIT2780
 1000 FORMAT(2(3I2,I3,1X))                                              EDIT2790
 2000 FORMAT(20A4)                                                      EDIT2800
 2001 FORMAT(13H  PLOT LIMITS/5X,7HMINIMUM,15F8.2/5X,7HMAXIMUM,15F8.2)  EDIT2810
 2002 FORMAT(/10H0 MANEUVER,I4,12H  START TIME,4I5,11H  STOP TIME,4I5)  EDIT2820
 2003 FORMAT(1H1,26X,20A4,13X,A10)                                      EDIT2830
 2004 FORMAT(/18H0 INPUT MATRICES :)                                    EDIT2840
```

```
2005 FORMAT(1H1,20X,20A4,10X,A1)/40X,14HNEWTON-RAPHSON,                    EDIT2850
     -     28H DIGITAL DERIVATIVE MATCHING/60X,10H1 APR 1974)              EDIT2860
2006 FORMAT(28H0D1 WILL BE DETERMINED USING,I3,                            EDIT2870
     -28H PASSES. RELAXATION FACTOR =,F5.2,13H  TOLERANCE =,F5.2)          EDIT2880
2007 FORMAT(46H THE A MATRIX INDICATES CASE IS LONGITUDINAL.               EDIT2890
     -57HLABELS ABOVE ARE WRONG. APPROPRIATE CORRECTIONS NOW MADE.)        EDIT2900
2008 FORMAT(23H0WMAPR WILL BE RUN WITH,I3,25H VALUES.   FIRST D., THEN,    EDIT2910
     -     E9.2,27H. THEREAFTER MULTIPLYING BY,E9.2)                       EDIT2920
2009 FORMAT( 50H0INPUT DATA (T INDICATES TRUE OR YES, F INDICATES ,        EDIT2930
     -     12HFALSE OR NO)/1H0,4X,3A4,5H CASE/16H     DATA SOURCE,         EDIT2940
     -     10H    CARD? ,L1,8X,6HTAPE? ,L1/5X,12HDATA RATE IS,F5.0,       EDIT2950
     -     57H SAMPLES/SECOND ON SOURCE FILE (IF 0, DETERMINED FROM TIM,EDIT2960
     -     22HES ON THE SOURCE FILE)/10X,26HDIVIDED BY THINNING FACTOR, EDIT2970
     -     3H OF,I3/5X,14HON INPUT TAPE#,I4,                               EDIT2980
     -     56H DATA WORDS PER RECORD.   SPECIAL SIGNAL ORDER DEFAULT? ,    EDIT2990
     -     L1)                                                             EDIT3000
2010 FORMAT(/16H0PROGRAM OPTIONS/24H0    APRIORI WEIGHTING =,E8.2,         EDIT3010
     -     13X,I3,22H TIME HALVINGS IN EAT./                               EDIT3020
     -     5X,12HITERATIONS =,I3,32H  (ITERATION WILL STOP IF ERROR ,      EDIT3030
     -     36HSUM CHANGES BY LESS THAN A FACTOR OF,E9.2,1H)/5X,            EDIT3040
     -     49HCASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN,E9.2)      EDIT3050
2011 FORMAT(/7H0OUTPUT/12H0    PLOTS? ,L1,25H  (NO PLOTS UNLESS FINAL ,EDIT3060
     -     22HERROR SUM IS LESS THAN,E9.3,1H)/10X,                         EDIT3070
     -     52HNUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED =,I3/     EDIT3080
     -     10X,24HSECONDS PER CENTIMETER =,F5.2/5X,                        EDIT3090
     -     50HPRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES? ,L1/5X,    EDIT3100
     -     57HEXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? ,EDIT3110
     -     L1/5X,51HPUNCHED FINAL NON-DIMENSIONAL DERIVATIVES AND CONFI,EDIT3120
     -     14HDENCE LEVELS? ,L1/5X,26HPUNCHED FINAL DIMENSIONAL ,          EDIT3130
     -     10HMATRICES? ,L1)                                               EDIT3140
2012 FORMAT(/54H0FLIGHT CONDITION AND VEHICLE CHACTERISTICS (0. INDICA,EDIT3150
     -     55HTES VALUE OBTAINED FROM TIME HISTORY ON QBAR,V OR MACH),     EDIT3160
     -     /44X,49H(MACH,ALPHA,CG AND PARAM ARE FOR REFERENCE ONLY, ,      EDIT3170
     -     20HNOT USED IN PROGRAM)/5X,14HMETRIC UNITS? ,L1/5X,             EDIT3180
     -     18HDYNAMIC PRESSURE =,F11.1,6X,10HVELOCITY =,F7.1/5X,           EDIT3190
     -     6HMACH =,F6.3,23X,7HALPHA =,F7.2,22H  (IF 999. , OBTAINED ,     EDIT3200
     -     18HFROM TIME HISTORY)/5X,19HCENTER OF GRAVITY =,F6.3,10X,       EDIT3210
     -     29HOTHER IDENTIFYING PARAMETER =,E10.3/5X,11HWING AREA =,       EDIT3220
     -     F7.1,17X,6HSPAN =,F7.2,5X,7HCHORD =,F6.2/5X,4HIX =,F9.1,22X,   EDIT3230
     -     4HIY =,F10.1,4X,4HIZ =,F10.1,4X,5HIXZ =,F8.1/5X,8HWEIGHT =,    EDIT3240
     -     F9.1/5X,26HINSTRUMENT OFFSETS FROM CG/                          EDIT3250
     -     10X,53HX-DIRECTION OFFSETS (+ = INSTRUMENT IS FORWARD OF CG)/EDIT3260
     -     14X,5HALPHA,F8.3,4H  AN,F8.3/14X,4HBETA,F9.3,4H  AY,F8.3/       EDIT3270
     -     10X,49HZ-DIRECTION OFFSETS ( + = INSTRUMENT IS BELOW CG)/       EDIT3280
     -     14X,4HBETA,F9.3,4H  AY,F8.3)                                    EDIT3290
2013 FORMAT(26H0SIGNAL SCALING AND BIASES/9H  SIGNALS,7X,14A8,A4/          EDIT3300
     -     10H  VAR BIAS,32X,3(7X,L1)/11H  VAR I.C. ,6X,L1,3(7X,L1)/       EDIT3310
     -     12H  FIXED BIAS,11F8.2/12H  SCALE FACT,7F8.2)                   EDIT3320
      END                                                                  EDIT3330
```

SUBROUTINE DATA

Description: Subroutine DATA reads the input time histories, performs any scaling and biasing required, and completes the program initialization. Averages of several time histories are obtained for use as default values for input parameters not set.

Programing notes: Comment cards separate major subroutine sections. If this is an intermediate step in the D1 determination or the *a priori* variation option, most of the subroutine is skipped since those sections were executed in the first step; this is true when the formal parameter IN is false.

Important variables –

X – vector containing one time point of the input time histories in degrees.

Z, DCR – vectors containing the input observations and controls in radians.

C – matrix containing factors for nondimensionalizing derivatives.

APR – matrix containing any off-diagonal *a priori* weightings. These weightings would be stored in the upper triangular portion of APR. There are no terms inserted here, but if such terms are desired, they may be inserted and the rest of the program will treat them properly. This matrix is referred to elsewhere in the program as SUM, and the lower triangular portion and the diagonal will be used to store other information.

APRD – vector containing the diagonal *a priori* weightings.

Subroutine listing:

```
      SUBROUTINE DATA(IN)                                                    DATA   0
C                                                                            DATA  10
C     READS TIME HISTORIES, PERFORMS VARIOUS INITIALIZATION                  DATA  20
C                                                                            DATA  30
      COMMON /ALLDIM/ MAX,MIX                                                DATA  40
      COMMON /COM/ NCASE,MZ,NPTS,NPTT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,        DATA  50
     -     D1TOL,D1RLX,NAPR,WFAC,WMAPR,ERRSUM,LAST,RATIO                     DATA  60
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                                  DATA  70
      COMMON /TOGIRL/JKM,JKMM,JKMM1,ERRMAX,ZEROIN,XT5,BOUND,APR,NI           DATA  80
     -     ,BIASK,ZERO,APRD,JKV,DIAG                                         DATA  90
      COMMON /INFO/ HH,NOITER,MX,MXP1,MU,TEST,XBN,ZBX,IPQR,IXYZ,XT3,MZM,     DATA 100
     -     CORECT,BIASKN,PLTMAX,XT4,ERRVEC,PUNCHD,NEAT                       DATA 110
      COMMON /DIMENS/ C,E,WQS,THETN                                          DATA 120
      COMMON /HEADNG/ LABELS,TITLE,JULIAN                                    DATA 130
      COMMON /TODATA/ S,SPAN,XB,ZB,XAY,CARD,ZAY,APRA,APRB,DCBIAS,            DATA 140
     -     XALF,THIN,TAPE,CBAR,APBP,STC,ETC,BIAS,AR,BR,XAN,ZAX,              DATA 150
     -     SCALE,NREC,ORDER,METRIC,AIX,AIY,AIZ,AIXZ,Q,V,GROSWT               DATA 160
      COMMON /ROUTH/ PUNCH ,PARAM,MACH,ALPHA,CG,AC,BC                        DATA 170
      DIMENSION A(5,4),B(5,8),TITLE(20),XT4(3),APRD(35),RECORD(100),         DATA 180
     -     AA(5,4),BB(5,8),X(15),Z(7),E(3,8), AP(4,4),BP(4,8),               DATA 190
     -     EXTRA(4), DCR(8),NPTS(15),D1(8,7),DC(4),                          DATA 200
     -     C(3,8) ,ERRVEC(20),APR(35,35),XT3(4),RI(5,4),                     DATA 210
     -     XT5(35),AR(5,4),BR(5,8),APRA(5,4),APRB(5,8),AC(5,4),BC(5,8)       DATA 220
      INTEGER T(4),ORDER(15),THIN,STC(15),ETC(15)                            DATA 230
      REAL DCBIAS(4),CALIB(7),BIAS(7),LABELS(15),MACH,SCALE(7)               DATA 240
      LOGICAL CORECT,METRIC,CARD,PLOTEM,APBP,AA,BB,TAPE,LAST,DIAG,           DATA 250
     -       BIASKN,ZEROIN,TEST,LONG,LATR,PRINT,IN,BIASK(3),ZERO(4),         DATA 260
     -     PUNCHD,PRNT                                                       DATA 270
      EQUIVALENCE(X(8),DC(1)),(X(12),EXTRA(1))                               DATA 280
      DATA STAR/1H*/,BLANK/1H /                                              DATA 290
      PRNT=PRINT.OR.(NOITER.EQ.0)                                            DATA 300
      G=32.172                                                               DATA 310
      IF(METRIC) G=9.80665                                                   DATA 320
      RAD=57.2958                                                            DATA 330
      LINE=50                                                                DATA 340
      DO 5 I=1,3                                                             DATA 350
    5 XT4(I)=0.                                                              DATA 360
      DO 10 I=1,7                                                            DATA 370
      CALIB(I)=1./RAD                                                        DATA 380
   10 Z(I)=0.                                                                DATA 390
      CALIB(5)=1.                                                            DATA 400
      DO 20 I=4,8                                                            DATA 410
   20 DCR(I)=0.                                                              DATA 420
      IF(.NOT.LONG) GO TO 50                                                 DATA 430
      XT4(1)=1.                                                              DATA 440
      CALIB(3)=1.                                                            DATA 450
      CALIB(7)=1.                                                            DATA 460
   50 CONTINUE                                                               DATA 470
C************************************** READ INPUT TIME HISTORY               DATA 480
      IF(.NOT.IN) GO TO 505                                                  DATA 490
      IF(PRNT) WRITE(3,2001)                                                 DATA 500
      NPTT=0                                                                 DATA 510
      AMACH=0.                                                               DATA 520
      ALFA=0.                                                                DATA 530
      AV=0.                                                                  DATA 540
      AQBAR=0.                                                               DATA 550
      PHI=0.                                                                 DATA 560
```

```
      THETA=0.                                                          DATA 570
      VEL=V                                                             DATA 580
      DO 500 I11=1,NCASE                                                DATA 590
      ISTMS=STC(I11)                                                    DATA 600
      IETMS=ETC(I11)                                                    DATA 610
      ITHIN=THIN-1                                                      DATA 620
      NPTS(I11)=0                                                       DATA 630
      IF(I11.LE.4) DCR(I11+4)=1.                                        DATA 640
  260 IF (TAPE) GO TO 240                                               DATA 650
      READ (1,1001) T,X                                                 DATA 660
      GO TO 250                                                         DATA 670
  240 READ (4) T,(RECORD(I),I=1,NREC)                                   DATA 680
  250 IF((T(4)+1000*(T(3)+60*T(2)+3600*T(1))).LT.ISTMS) GO TO 260       DATA 690
      ITST=T(4)                                                         DATA 700
      IF(.NOT.TAPE) GO TO 300                                           DATA 710
      DO 270 I=1,15                                                     DATA 720
  270 X(I)=RECORD(ORDER(I))                                             DATA 730
  300 ITM=T(4)+1000*(T(3)+60*T(2)+3600*T(1))                            DATA 740
      IF(ITM.GT.IETMS) GO TO 430                                        DATA 750
      ITHIN=ITHIN+1                                                     DATA 760
      IF(MOD(ITHIN,THIN),NE.0) GO TO 385                                DATA 770
          NPTS(I11) = NPTS(I11) + 1                                     DATA 780
      IF(NPTS(1).NE. 2.OR. SPS.NE.0.) GO TO 309                         DATA 790
      I=T(4)-ITST                                                       DATA 800
      IF(I.LT.0) I=I+1000                                               DATA 810
      HH=((I+2)/5)*5                                                    DATA 820
      HH=HH/1000.                                                       DATA 830
      SPS=1./HH                                                         DATA 840
C************************************* ADD BIASES AND SCALE FACTORS      DATA 850
  309 DO 310 I=1,4                                                      DATA 860
      DC(I)=DC(I)+DCBIAS(I)                                             DATA 870
  310 DCR(I)=DC(I)/RAD                                                  DATA 880
      DO 315 I=1,MZ                                                     DATA 890
  315 X(I)=X(I)*SCALE(I)+BIAS(I)                                        DATA 900
      AMACH=AMACH+EXTRA(3)                                              DATA 910
      AQBAR=AQBAR+EXTRA(4)                                              DATA 920
      IF (LONG) GO TO 350                                               DATA 930
      IF(.NOT.CORECT) GO TO 340                                         DATA 940
      IF( V.EQ.0.) VEL=EXTRA(2)                                         DATA 950
      X(1)=X(1)-(XB*X(3)-ZB*X(2))/VEL                                   DATA 960
  340 ALFA=ALFA+EXTRA(1)                                                DATA 970
      AV=AV+EXTRA(2)                                                    DATA 980
      GO TO 360                                                         DATA 990
  350 IF(.NOT.CORECT)  GO TO 355                                        DATA1000
      IF( V.EQ.0.) VEL=X(3)                                             DATA1010
      X(1)=X(1)+XALF*X(2)/VEL                                           DATA1020
  355 ALFA=ALFA+X(1)                                                    DATA1030
      AV=AV+X(3)                                                        DATA1040
      PHI=PHI+EXTRA(1)                                                  DATA1050
      THETA=THETA+X(4)                                                  DATA1060
  360 DO 365 I=1,MZ                                                     DATA1070
  365 Z(I)=X(I)*CALIB(I)                                                DATA1080
      ITIME=T(4)+1000*(T(3)+100*T(2)+10000*T(1))                        DATA1090
      WRITE (7) ITIME,Z,DCR,EXTRA                                       DATA1100
      IF(.NOT.PRNT) GO TO 375                                           DATA1110
      IF(MOD(LINE,50).EQ.0)                                             DATA1120
     -WRITE(3,2003)  TITLE,JULIAN,GROSWT,AIX,AIZ,AIXZ,AIY,Q,V,LABELS    DATA1130
```

```
      WRITE(3,2010)T,X                                                DATA1140
      LINE=LINE+1                                                     DATA1150
  375 IF(ITM.EQ.IETMS) GO TO 430                                      DATA1160
  385 IF (CARD) GO TO 400                                             DATA1170
      READ (4) T,(RECORD(I),I=1,NREC)                                 DATA1180
      DO 390 I=1,15                                                   DATA1190
  390 X(I)=RECORD(ORDER(I))                                           DATA1200
      GO TO 300                                                       DATA1210
  400 READ (1,1001) T,X                                               DATA1220
      GO TO 300                                                       DATA1230
  430 IF(NPTS(I11).GT.0) GO TO 435                                    DATA1240
      WRITE(3,2000)I11                                                DATA1250
      STOP                                                            DATA1260
  435 NPTT=NPTT+NPTS(I11)                                             DATA1270
      WRITE(3,2007)I11,NPTS(I11)                                      DATA1280
  500 CONTINUE                                                        DATA1290
      ANPT=FLOAT(NPTT)                                                DATA1300
      IF(MACH.EQ.0.) MACH=AMACH/ANPT                                  DATA1310
      IF(ALPHA.EQ.999.) ALPHA=ALFA/ANPT                               DATA1320
      IF(V.EQ.0.) V=AV/ANPT                                           DATA1330
      IF(Q.EQ.0.) Q=AQBAR/ANPT                                        DATA1340
      VOG=V/G                                                         DATA1350
      AM=GROSWT*VOG/(Q*S)                                             DATA1360
      V2=2.*V                                                         DATA1370
      IF (LONG) GO TO 170                                             DATA1380
C************************************* LATERAL SETUP                 DATA1390
      XAN=-ZAY                                                        DATA1400
      ZAX=XAY                                                         DATA1410
      IPQR=3                                                          DATA1420
      IXYZ=1                                                          DATA1430
      P1=VOG                                                          DATA1440
      P3=1.                                                           DATA1450
      AP3=1.                                                          DATA1460
      QSB=Q*S*SPAN                                                    DATA1470
      QSBB=QSB*SPAN                                                   DATA1480
      C(1,2)=1.                                                       DATA1490
      C(2,2)=V2*AIX/QSBB                                              DATA1500
      C(3,2)=V2*AIZ/QSBB                                              DATA1510
      C(1,3)=0.                                                       DATA1520
      C(2,3)=C(2,2)                                                   DATA1530
      C(3,3)=C(3,2)                                                   DATA1540
      C(1,1)=AM/RAD                                                   DATA1550
      C(2,1)=AIX/(QSB*RAD)                                            DATA1560
      C(3,1)=AIZ/(QSB*RAD)                                            DATA1570
      DO 160 I=1,3                                                    DATA1580
      C(I,8)=C(I,1)*RAD                                               DATA1590
      DO 160 J=4,7                                                    DATA1600
  160 C(I,J)=C(I,1)                                                   DATA1610
      GO TO 200                                                       DATA1620
C************************************* LONGITUDINAL SETUP            DATA1630
  170 QSCI=Q*S*CBAR/AIY                                               DATA1640
      THETA=THETA/ANPT                                                DATA1650
      WQS=COS(THETA/RAD)*COS(PHI/(RAD*ANPT))*GROSWT/(Q*S)             DATA1660
      IPQR=2                                                          DATA1670
      IXYZ=3                                                          DATA1680
      P1=-VOG                                                         DATA1690
      P3=1./G                                                         DATA1700
```

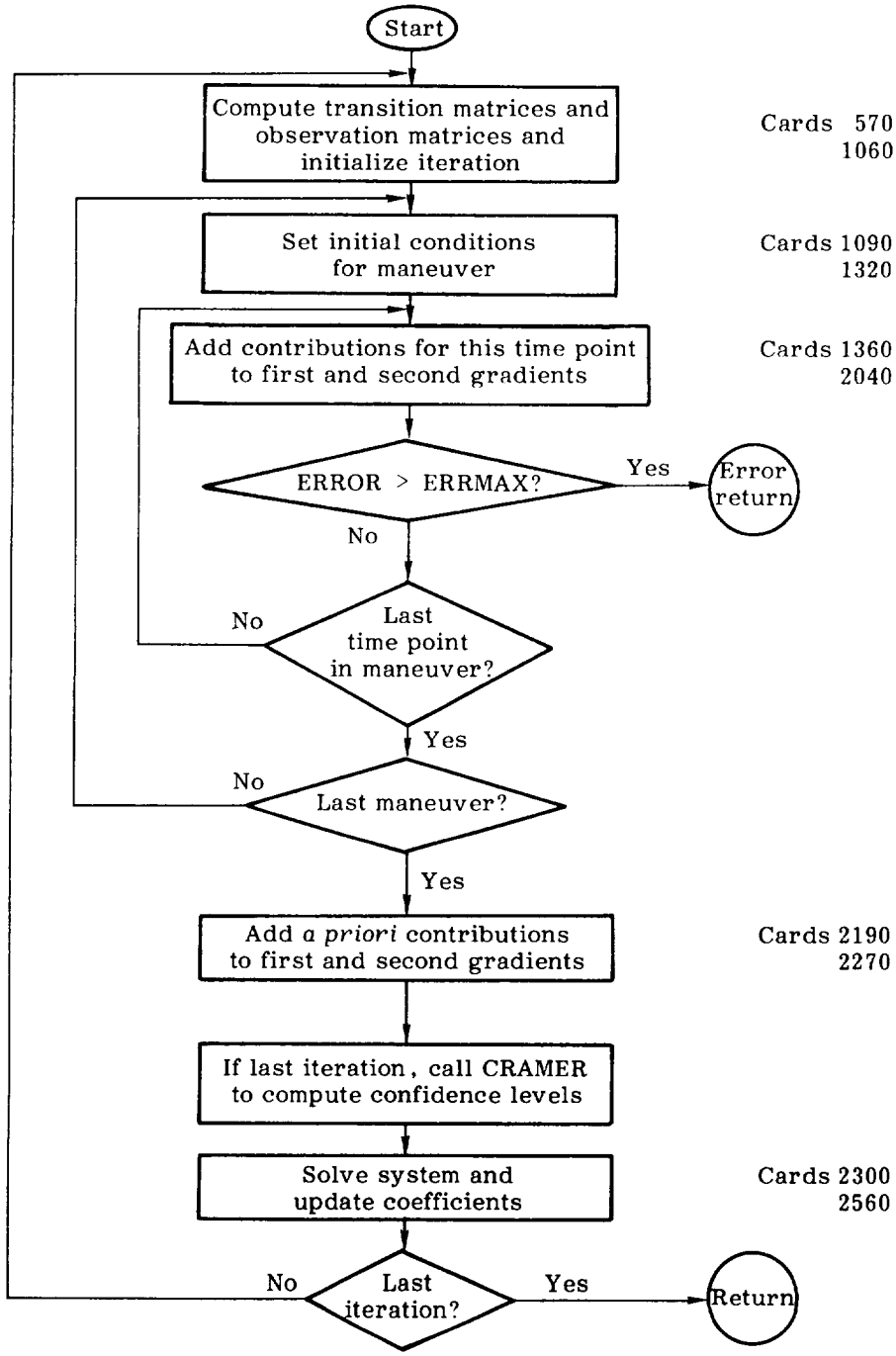
```
      AP3=0.                                                            DATA1710
      C(1,1)=AM/RAD                                                     DATA1720
      C(2,1)=1./(QSCI*RAD)                                              DATA1730
      C(3,1)=C(1,1)/V                                                   DATA1740
      C(1,2)=0.                                                         DATA1750
      C(2,2)=V2/(QSCI*CBAR)                                             DATA1760
      C(3,2)=C(1,2)                                                     DATA1770
      DO 180 I=1,3                                                      DATA1780
      C(I,8)=C(I,1)*RAD                                                 DATA1790
      C(I,3)=C(I,8)*V/2.                                                DATA1800
      DO 180 J=4,7                                                      DATA1810
  180 C(I,J)=C(I,1)                                                     DATA1820
      THETN=THETA*C(1,1)                                                DATA1830
C     SET E=00*00 IF DERIVATIVE IS FIXED, OTHERWISE E=00 00             DATA1840
  200 DO 220 I=1,3                                                      DATA1850
       DO 210 J=1,3                                                     DATA1860
      E(I,J)=STAR                                                       DATA1870
      IF(.NOT.AA(I,J)    ) E(I,J)=BLANK                                 DATA1880
  210 CONTINUE                                                          DATA1890
      DO 220 J=4,8                                                      DATA1900
      K=J-3                                                             DATA1910
      E(I,J)=STAR                                                       DATA1920
      IF(.NOT.BB(I,K)    ) E(I,J)=BLANK                                 DATA1930
  220 CONTINUE                                                          DATA1940
      MAX=8                                                             DATA1950
C************************************* FORM AP AND BP IF NOT READ IN     DATA1960
      IF (APBP) GO TO 129                                               DATA1970
      DO 112 J=1,MU                                                     DATA1980
      BP(1,J)=P1                                                        DATA1990
      BP(2,J)=1.                                                        DATA2000
  112 BP(3,J)=P3                                                        DATA2010
      DO 114 J=1,MX                                                     DATA2020
      AP(1,J)=0.                                                        DATA2030
      AP(2,J)=1.                                                        DATA2040
  114 AP(3,J)=AP3                                                       DATA2050
      AP(1,1)=P1                                                        DATA2060
      AP(3,1)=P3                                                        DATA2070
  129 CONTINUE                                                          DATA2080
      XBN=XAN/G                                                         DATA2090
      ZBX=ZAX/G                                                         DATA2100
C************************************* STORE APRIORI WEIGHTINGS          DATA2110
  505 MAX=35                                                            DATA2120
      CALL AZOT(APR)                                                    DATA2130
      DO 510 I=1,35                                                     DATA2140
      APRD(I)=0.                                                        DATA2150
  510 XT5(I)=0.                                                         DATA2160
      IF(WMAPR.EQ.0.) RETURN                                            DATA2170
      K=0                                                               DATA2180
      DO 525 I=1,MX                                                     DATA2190
      DO 520 J=1,MU                                                     DATA2200
      IF(BB(I,J)       ) GO TO 520                                      DATA2210
      K=K+1                                                             DATA2220
      XT5(K)=B(I,J)-BR(I,J)                                             DATA2230
      APRD(K)=APRB(I,J)*WMAPR                                           DATA2240
  520 CONTINUE                                                          DATA2250
      DO 525 J=1,MX                                                     DATA2260
      IF(AA(I,J)       ) GO TO 525                                      DATA2270
```

```
      K=K+1                                                              DATA2280
      XT5(K)=A(I,J)-AR(I,J)                                              DATA2290
      APRD(K)=APRA(I,J)*WMAPR                                            DATA2300
  525 CONTINUE                                                           DATA2310
 1001 FORMAT(3I2,I4,7F10.4/8F10.4)                                       DATA2320
 2000 FORMAT(14HOTIME INTERVAL,I3,10H NOT FOUND)                         DATA2330
 2001 FORMAT(55HOINPUT TIME HISTORY WITH BIASES AND SCALE FACTORS APPLI,DATA2340
     -      38HED AND VANE CORRECTIONS ADDED FOLLOWS.)                  DATA2350
 2003 FORMAT(1H1,26X,20A4,13X,A10/4HOW =,F8.1,6H  IX =,F9.1,6H  IZ =,    DATA2360
     -      F10.1,7H  IXZ =,F7.1,6H  IY =,F9.1,8H  QBAR =,F7.2,5H  V =, DATA2370
     -      F8.2/5X,4HTIME,6X,14A8,A4)                                  DATA2380
 2007 FORMAT(1H0,40X,35HTOTAL NUMBER OF POINTS FOR MANEUVER,I3,2H =,I6)  DATA2390
 2010 FORMAT(1X,3I2,I3,2X,12F8.3,F8.1,F8.3,F8.2)                         DATA2400
      RETURN                                                             DATA2410
      END                                                                DATA2420
```

SUBROUTINE AGIRL

Description: Subroutine AGIRL performs the parameter estimation. Almost all the routine is skipped if NOITER = 0.

Flow chart:



Programing notes: For derivation of the form of the first and second gradients, see reference 3.

Important variables –

SUM – contains second gradient in lower triangular and diagonal locations, and off-diagonal *a priori* weightings in upper triangular. Diagonal *a priori* weightings are in APRD. The first gradient appears as an extra column in SUM (the JKMth column). The SUM matrix is printed each iteration when TEST = T.

XJI – $\nabla_c (z_i - y_i)^*$

RIA – $R^{-1}A$

RIB – $R^{-1}B$

PHI1 – $e^{R^{-1}A\Delta t}$

APHI – $\left(\int_0^{\Delta t} e^{R^{-1}A\tau} d\tau\right) R^{-1}$

BPHI – (APHI)(B)

AAP, BBP – observation matrices formed from A and AP or B and BP, with any terms for accelerometer offset from the center of gravity added. (These matrices are referred to as G and H in the derivation.)

RIAP – array of partial derivatives of AAP with respect to A.

$$\text{RIAP(I,J,K)} = \frac{\partial \text{AAP(I,K)}}{\partial \text{A(J,K)}}.$$

RIBP – array of partial derivatives of BBP with respect to B.

$$\text{RIBP(I,J,K)} = \frac{\partial \text{BBP(I,K)}}{\partial \text{B(J,K)}}.$$

Z, U – measured values of observations and controls.

XT1, XT2 – computed values for observations.

XT3 – variable initial conditions on the states.

XT4 – variable bias on the observations other than states.

XT5 – difference between the estimated coefficients and the *a priori* values.

PB – solution vector for the change in the estimates of the coefficients.

MX – number of states.

MZ – number of observations.

Subroutine listing:

```
      SUBROUTINE AGIRL                                                  AGIR   0
C                                                                       AGIR  10
C     CORE SUBROUTINE - ITERATIVE LOOP                                  AGIR  20
C                                                                       AGIR  30
      COMMON /ALLDIM/ MAX,MIX                                           AGIR  40
      COMMON /COM/ NCASE,MZ,NPTS,NPTT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,   AGIR  50
     -     D1TOL,D1RLX,NAPR,WFAC,WMAPR,ERRSUM,LAST,RATIO                AGIR  60
      COMMON /TOGIRL/JKM,JKMM,JKMM1,ERRMAX,ZEROIN,XT5,BOUND,SUM,NI      AGIR  70
     -     ,BIASK,ZERO,APRO,JKV,DIAG                                    AGIR  80
      COMMON /INFO/ HH,NOITER,MX,MXP1,MU,TEST,XAN,ZAX,IPQR,IXYZ,XT3,MZM,AGIR  90
     -     CORECT,BIASKN,PLTMAX,XT4,ERRVEC,PUNCHD,NEAT                  AGIR 100
      COMMON /HEADNG/  LABELS,TITLE,JULIAN                              AGIR 110
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                             AGIR 120
      COMMON /ROUTH/ PUNCH,PARAM,MACH,ALP,CG,AC,BC                      AGIR 130
      COMMON /DIMENS/ C,CC,E,WQS,THETN                                  AGIR 140
      REAL XT3(4),XT4(3),XT5(35),PB(35), APRO(35),LABELS(15)            AGIR 150
      REAL U12(8),U23(8),BPHI(5,8),EXTRA(4),AC(5,4),BC(5,8)             AGIR 160
      LOGICAL AA,BB,TEST,ZEROIN,BIASKN,CORECT,LATR,ERSTOP,PUNCHD        AGIR 170
     -     ,ZERO(4),BIASK(3),PRINT,LAST,DIAG,E(24),PUNCH,PLOTEM         AGIR 180
      DIMENSION A( 5,4 ), B( 5,8 ), SUM(35,35),PHI1( 5,4 ), U( 8,3 ),  AGIR 190
     -          AA( 5,4 ),BB( 5,8 ),RI(5,4),     APHI( 5,4 ), Z( 8,3 ),AGIR 200
     -          AP( 4,4 ),BP( 4,8 ), XJI(35,8 ),DUM1( 5,4 ),D1( 8,7 ),AGIR 210
     -                    XT12(4),AAP(3,4),BBP(3,8),RIA(5,4),RIB(5,8),  AGIR 220
     -      XT1(7),NPTS(15),ERRVEC(20),XT2(7),D2(8),TITLE(20),XTE(4),   AGIR 230
     -     RIBP(4,4,8),RIAP(4,4,4),XJID1(35,7),C(3,3),CC(3,5)           AGIR 240
      EQUIVALENCE (APHI(5,3),APHIL),(PHI1(5,3),PHI1L),(BPHI(5,3),BPHIL) AGIR 250
      DATA PHI1L/4HPHI1/,APHIL/4HAPHI/,BPHIL/4HPHIB/                    AGIR 260
      ANPT=FLOAT(NPTT)                                                  AGIR 270
      ERMX=ERRMAX*ANPT                                                  AGIR 280
      EN=1.E+50                                                         AGIR 290
      ERSTOP=.FALSE.                                                    AGIR 300
      DO 7 I=1,JKMM1                                                    AGIR 310
    7 PB(I)=0.                                                          AGIR 320
      DO 2 I = 1,MX                                                     AGIR 330
    2 XT3(I)=0.                                                         AGIR 340
      IF (NOITER.EQ.0) GO TO 600                                        AGIR 350
      WRITE(3,1000)TITLE,JULIAN,MACH,ALP,PARAM,CG                       AGIR 360
      CALL DERIV(LONG)                                                  AGIR 370
      DO 320 I=1,MX                                                     AGIR 380
      DO 320 J=1,MX                                                     AGIR 390
      DO 310 K=1,MU                                                     AGIR 400
  310 RIBP(I,J,K)=RI(I,J)*BP(I,K)                                       AGIR 410
      DO 320 K=1,MX                                                     AGIR 420
  320 RIAP(I,J,K)=RI(I,J)*AP(I,K)                                       AGIR 430
C     TERMS FOR ACCELEROMETER OFFSET FROM CG                            AGIR 440
      IF(.NOT.CORECT) GO TO 350                                         AGIR 450
      DO 340 J=1,MX                                                     AGIR 460
      DO 330 K=1,MU                                                     AGIR 470
      RIBP(1,J,K)=RIBP(1,J,K)+XAN*RI(2,J)                               AGIR 480
  330 RIBP(IXYZ,J,K)=RIBP(IXYZ,J,K)+ZAX*RI(IPQR,J)                      AGIR 490
      DO 340 K=1,MX                                                     AGIR 500
      RIAP(1,J,K)=RIAP(1,J,K)+XAN*RI(2,J)                               AGIR 510
  340 RIAP(IXYZ,J,K)=RIAP(IXYZ,J,K)+ZAX*RI(IPQR,J)                      AGIR 520
  350 CONTINUE                                                          AGIR 530
      WRITE(3,103)JKMM1                                                 AGIR 540
C*************************************  ITERATION LOOP                   AGIR 550
      DO 32 LL = 1,NOITER                                               AGIR 560
```

```
      MAX    = 5                                                        AGIR 570
      CALL ASPIT(A)                                                     AGIR 580
      CALL ASPIT(B)                                                     AGIR 590
      CALL AMULT(RI,A,RIA)                                              AGIR 600
      CALL AMULT(RI,E,RIB)                                              AGIR 610
C     COMPUTE A*AP AND B*BP          ADD TERMS FOR CG SHIFT IF NEEDED   AGIR 620
      DO 45 I=1,3                                                       AGIR 630
      DO 40 J=1,MX                                                      AGIR 640
   40 AAP(I,J)=RIA(I,J)*AP(I,J)                                         AGIR 650
      DO 45 J=1,MU                                                      AGIR 660
   45 BBP(I,J)=RIB(I,J)*BP(I,J)                                         AGIR 670
      IF(.NOT.CORECT) GO TO 50                                          AGIR 680
      DO 46 J=1,MX                                                      AGIR 690
      AAP(1,J)=AAP(1,J)+XAN*RIA(2,J)                                    AGIR 700
   46 AAP(IXYZ,J)=AAP(IXYZ,J)+ZAX*RIA(IPQR,J)                           AGIR 710
      DO 47 J=1,MU                                                      AGIR 720
      BBP(1,J)=BBP(1,J)+XAN*RIB(2,J)                                    AGIR 730
   47 BBP(IXYZ,J)=BBP(IXYZ,J)+ZAX*RIB(IPQR,J)                           AGIR 740
   50 REWIND 7                                                          AGIR 750
      CALL AEAT(RIA,HH,PHI1,APHI,DUM1,SUM,NEAT)                         AGIR 760
      CALL AMULT(APHI,RI,DUM1)                                          AGIR 770
      CALL AMAKE(APHI,DUM1)                                             AGIR 780
      CALL AMULT(APHI,B,BPHI)                                           AGIR 790
      IF(.NOT.TEST) GO TO 51                                            AGIR 800
      CALL ASPIT(PHI1)                                                  AGIR 810
      CALL ASPIT(APHI)                                                  AGIR 820
      CALL ASPIT(BPHI)                                                  AGIR 830
   51 DO 53 I=1,MX                                                      AGIR 840
      DO 53 J=1,I                                                       AGIR 850
      TEMP=PHI1(I,J)                                                    AGIR 860
      PHI1(I,J)=PHI1(J,I)                                               AGIR 870
   53 PHI1(J,I)=TEMP                                                    AGIR 880
      MAX    = NI                                                       AGIR 890
      DO 60 I=1,JKM                                                     AGIR 900
      DO 60 J=1,I                                                       AGIR 910
   60 SUM(I,J)=0.                                                       AGIR 920
      DO 52 I=1,8                                                       AGIR 930
   52 D2(I) = 0.0                                                       AGIR 940
C     VARIABLE BIAS                                                     AGIR 950
      IF(.NOT.BIASKN) GO TO 3                                           AGIR 960
      IBIAS=JKV                                                         AGIR 970
      DO 16 I = 1,MZM                                                   AGIR 980
      IF(.NOT.BIASK(I)) GO TO 16                                        AGIR 990
      IBIAS  = IBIAS + 1                                                AGIR1000
      DO 15 J = 1,MZ                                                    AGIR1010
   15 XJI(IBIAS,J)=0.                                                   AGIR1020
      XJI(IBIAS,I+MX)=1.                                                AGIR1030
   16 CONTINUE                                                          AGIR1040
      WRITE(3,1001)(LABELS(I),I=MXP1,MZ)                                AGIR1050
      WRITE(3,102)(XT4(I),I=1,MZM)                                      AGIR1060
C************************************  CASE LOOP                        AGIR1070
    3 DO 26 LM = 1,NCASE                                                AGIR1080
      NNM1=NPTS(LM)-1                                                   AGIR1090
      XJI(NI,1)=JKV                                                     AGIR1100
      XJI(NI,2)=MX                                                      AGIR1110
      CALL AZOT(XJI)                                                    AGIR1120
      READ (7) IT,XT1,(U(K,1),K=1,8),EXTRA                              AGIR1130
```

```
      READ (7) IT,XT2,(U(K,2),K=1,8),EXTRA                              AGIR1140
C     VARIABLE INITIAL CONDITION                                        AGIR1150
      IF(.NOT.ZEROIN) GO TO 6                                           AGIR1160
      IC=JKMM                                                           AGIR1170
      DO 4 I = 1,MX                                                     AGIR1180
      IF(.NOT.ZERO(I)) GO TO 4                                          AGIR1190
      IC     = IC + 1                                                   AGIR1200
      XJI(IC,I)=1.                                                      AGIR1210
      XT1(I) = XT1(I) + XT3(I)                                          AGIR1220
      XT2(I) = XT2(I) + XT3(I)                                          AGIR1230
    4 CONTINUE                                                          AGIR1240
      IF(LM.NE.1) GO TO 6                                               AGIR1250
      WRITE(3,1001)(LABELS(I),I=1,MX)                                   AGIR1260
      WRITE(3,108) (XT3(I),I=1,MX)                                      AGIR1270
    6 DO 8 I=1,MZ                                                       AGIR1280
      Z(I,1) = XT1(I)                                                   AGIR1290
    8 Z(I,2) = XT2(I)                                                   AGIR1300
      IF(TEST) WRITE(3,111)(XT1(I),I=1,MZ)                              AGIR1310
      IF(TEST) WRITE(3,111)(XT2(I),I=1,MZ)                              AGIR1320
C************************************  TIME LOOP                         AGIR1330
C************************************  COMPUTE GRADIENT AND HESSIAN      AGIR1340
      DO 10025 IP = 2,NNM1                                              AGIR1350
      READ (7) IT,(Z(K,3),K=1,7),(U(J,3),J=1,8),EXTRA                   AGIR1360
      IF(LL.EQ.1) GO TO 203                                             AGIR1370
      DO 201 I=1,MX                                                     AGIR1380
      XT12(I)=.5*(XT1(I)+XT2(I))                                        AGIR1390
  201 XT6(I)=XT2(I)                                                     AGIR1400
      GO TO 205                                                         AGIR1410
  203 DO 204 I=1,MX                                                     AGIR1420
      XT12(I)=.5*(Z(I,2)+Z(I,1))                                        AGIR1430
      XT6(I)=Z(I,2)                                                     AGIR1440
      Z(I,1) = Z(I,2)                                                   AGIR1450
  204 Z(I,2) = Z(I,3)                                                   AGIR1460
  205 CONTINUE                                                          AGIR1470
      DO 206 I=1,MU                                                     AGIR1480
      U12(I)=.5*(U(I,1)+U(I,2))                                         AGIR1490
      U23(I)=.5*(U(I,2)+U(I,3))                                         AGIR1500
      U(I,1)=U(I,2)                                                     AGIR1510
  206 U(I,2)=U(I,3)                                                     AGIR1520
      DO 210 J=MXP1,MZ                                                  AGIR1530
      DO 210 JK=1,JKV                                                   AGIR1540
  210 XJI(JK,J) = 0.0                                                   AGIR1550
      DO 11 I = 1,MX                                                    AGIR1560
      XT1(I)=XT2(I)                                                     AGIR1570
   11 XT2(I)=0.                                                         AGIR1580
      CALL AMULT(XJI,PHI1,XJID1)                                        AGIR1590
      CALL AMAKE(XJI,XJID1)                                             AGIR1600
      JK     = 0                                                        AGIR1610
      DO 14 J = 1,MX                                                    AGIR1620
      DO 12 K = 1,MU                                                    AGIR1630
      XT2(J)=XT2(J)+BPHI(J,K)*U23(K)                                    AGIR1640
      IF (BB(J,K)) GO TO 12                                             AGIR1650
      JK     = JK + 1                                                   AGIR1660
      DO 115 I=1,MX                                                     AGIR1670
      XJI(JK,I+MX)=RIBP(I,J,K)*U(K,1)                                   AGIR1680
  115 XJI(JK,I)=XJI(JK,I)+U12(K)*APHI(I,J)                              AGIR1690
   12 CONTINUE                                                          AGIR1700
```

```
      DO 14 K = 1,MX                                                    AGIR1710
      XT2(J)=XT2(J)+PHI1(K,J)*XT1(K)                                    AGIR1720
      IF(AA(J,K)) GO TO 14                                              AGIR1730
      JK      = JK + 1                                                  AGIR1740
      DO 125 I=1,MX                                                     AGIR1750
      XJI(JK,I+MX)=PIAP(I,J,K)*XT6(K)                                   AGIR1760
  125 XJI(JK,I)=XJI(JK,I)+XT12(K)*APHI(I,J)                             AGIR1770
   14 CONTINUE                                                          AGIR1780
      DO 19 L = MXP1,MZ                                                 AGIR1790
      LMMX=L-MX                                                         AGIR1800
      DO 17 JK=1,JKV                                                    AGIR1810
      DO 17 K = 1,MX                                                    AGIR1820
   17 XJI(JK,L)=XJI(JK,L) + XJI(JK,K)*AAP(LMMX,K)                       AGIR1830
      XT2(L)=XT4(LMMX)                                                  AGIR1840
      DO 18 K = 1,MU                                                    AGIR1850
   18 XT2(L)=XT2(L)+BBP(LMMX,K)*U(K,2)                                  AGIR1860
      DO 19 K = 1,MX                                                    AGIR1870
   19 XT2(L)=XT2(L)+AAP(LMMX,K)*XT2(K)                                  AGIR1880
      DO 20 J = 1,MZ                                                    AGIR1890
      XJI(JKM,J) = Z(J,3) - XT2(J)                                      AGIR1900
   20 D2(J)      = XJI(JKM,J)**2 + D2(J)                                AGIR1910
      IF(TEST) WRITE(3,111)(XT2(I),I=1,MZ)                              AGIR1920
      IF(DIAG) GO TO 62                                                 AGIR1930
      MIX=8                                                             AGIR1940
      XJI(NI,1)=JKM                                                     AGIR1950
      XJI(NI,2)=MZ                                                      AGIR1960
      CALL AMULT(XJI,D1,XJID1)                                          AGIR1970
      XJI(NI,1)=JKV                                                     AGIR1980
      XJI(NI,2)=MX                                                      AGIR1990
      MIX=5                                                             AGIR2000
      GO TO 63                                                          AGIR2010
   62 CALL DMULT(XJI,D1,XJID1,JKM,MZ)                                   AGIR2020
   63 CALL SUMULT(XJI,XJID1,SUM,JKM,MZ)                                 AGIR2030
      IF(SUM(JKM,JKM).GT.ERMX) GO TO 510                                AGIR2040
10025 CONTINUE                                                          AGIR2050
   26 CONTINUE                                                          AGIR2060
C************************************* END OF TIME AND CASE LOOPS       AGIR2070
      ERRSUM=SUM(JKM,JKM)/ANPT                                          AGIR2080
      ERRVEC(LL)=ERRSUM                                                 AGIR2090
      WRITE(3,104)ERRSUM                                                AGIR2100
      IF(ABS((EN-ERRSUM)/EN).LT.BOUND) ERSTOP=.TRUE.                    AGIR2110
      EN=ERRSUM                                                         AGIR2120
      DO 64 I=1,MZ                                                      AGIR2130
      XT1(I)=D2(I)/ANPT                                                 AGIR2140
   64 D2(I)=XT1(I)*D1(I,I)                                              AGIR2150
      WRITE(3,105)(XT1(I),I=1,MZ)                                       AGIR216
      WRITE(3,106)(D2(I),I=1,MZ)                                        AGIR2170
C************************************* SOLUTION OF SYSTEM               AGIR2180
      DO 28 I =1,JKMM1                                                  AGIR2190
      XT5(I) = XT5(I) + PB(I)                                           AGIR2200
      SUM(I,JKM)=SUM(JKM,I)-XT5(I)*APRD(I)                              AGIR2210
      SUM(I,I)=SUM(I,I)+APRD(I)                                         AGIR2220
      IM1=I-1                                                           AGIR2230
      IF(IM1.EQ.0) GO TO 28                                             AGIR2240
      DO 27 J=1,IM1                                                     AGIR2250
   27 SUM(I,J)=SUM(I,J)+SUM(J,I)                                        AGIR2260
   28 CONTINUE                                                          AGIR2270
```

```
      IF (TEST) CALL ASPIT(SUM)                                          AGIR2280
      IF(ERSTOP.OR.(LL.EQ.NOITER)) CALL CRAMER(SUM,APRO,MU,MZ,ERRSUM)    AGIR2290
      CALL SOLVE(SUM,PB)                                                 AGIR2300
      IF(TEST) WRITE(3,107)(PB(I),I=1,JKMM1)                             AGIR2310
C************************************** UPDATE COEFFICIENTS               AGIR2320
      IJ              = 0                                                AGIR2330
      DO 31 I = 1,MX                                                     AGIR2340
      DO 30 J = 1,MU                                                     AGIR2350
      IF (BB(I,J)       ) GO TO 30                                       AGIR2360
      IJ              = IJ + 1                                           AGIR2370
      B(I,J)          = PB(IJ) + B(I,J)                                  AGIR2380
   30 CONTINUE                                                           AGIR2390
      DO 31 J = 1,MX                                                     AGIR2400
      IF (AA(I,J)       ) GO TO 31                                       AGIR2410
      IJ              = IJ + 1                                           AGIR2420
      A(I,J)          = PB(IJ) + A(I,J)                                  AGIR2430
   31 CONTINUE                                                           AGIR2440
      IF(.NOT.ZEROIN) GO TO 35                                           AGIR2450
      DO 34 I=1,MX                                                       AGIR2460
      IF(.NOT.ZERO(I)) GO TO 34                                          AGIR2470
      IJ=IJ+1                                                            AGIR2480
      XT3(I)=XT3(I)+PB(IJ)                                               AGIR2490
   34 CONTINUE                                                           AGIR2500
   35 IF (.NOT.BIASKN ) GO TO 37                                         AGIR2510
      DO 36 I=1,MZM                                                      AGIR2520
      IF (.NOT.BIASK(I)) GO TO 36                                        AGIR2530
      IJ=IJ+1                                                            AGIR2540
      XT4(I) = XT4(I) + PB(IJ)                                           AGIR2550
   36 CONTINUE                                                           AGIR2560
   37 WRITE(3,101)LL                                                     AGIR2570
      IF (ERSTOP) GO TO 38                                               AGIR2580
   32 CONTINUE                                                           AGIR2590
C************************************** END OF ITERATION LOOP             AGIR2600
      GO TO 500                                                          AGIR2610
   38 WRITE(3,110)BOUND                                                  AGIR2620
      NOITER=LL                                                          AGIR2630
  500 MAX=5                                                              AGIR2640
      WRITE(3,2003)                                                      AGIR2650
      CALL ASPIT(AC)                                                     AGIR2660
      CALL ASPIT(BC)                                                     AGIR2670
      WRITE(3,2006)                                                      AGIR2680
      DO 508 I=1,3                                                       AGIR2690
      DO 507 J=1,3                                                       AGIR2700
  507 AC(I,J)=AC(I,J)*C(I,J)                                             AGIR2710
      DO 508 J=1,5                                                       AGIR2720
  508 BC(I,J)=BC(I,J)*CC(I,J)                                            AGIR2730
      CALL ASPIT(AC)                                                     AGIR2740
      CALL ASPIT(BC)                                                     AGIR2750
      RETURN                                                             AGIR2760
  510 WRITE(3,2001)ERRMAX                                                AGIR2770
      NOITER=LL                                                          AGIR2780
      ERRVEC(LL)=ERRMAX                                                  AGIR2790
  101 FORMAT(//50X,16HITERATION NUMBER,I4,10H COMPLETED)                 AGIR2800
  102 FORMAT(15H VARIABLE BIAS ,3E12.4)                                  AGIR2810
  103 FORMAT(1H+,100X,20HNUMBER OF UNKNOWNS =,I3/1HO,20X,                AGIR2820
     -      23HENTERING ITERATION LOOP/25HO DIMENSIONAL DERIVATIVE ,     AGIR2830
     -      39HMATRICES PER RADIAN, BIASES IN RADIANS.//)                AGIR2840
```

```
 104 FORMAT(95X,20HWEIGHTED ERROR SUM =,E12.4)                          AGIR2850
 105 FORMAT(7H ERRORS/1X,11E12.4)                                       AGIR2860
 106 FORMAT(16H WEIGHTED ERRORS/1X,11E12.4)                             AGIR2870
 107 FORMAT(12H PB VECTOR =,10E12.4/(12X,10E12.4))                      AGIR2880
 108 FORMAT(15H VARIABLE IC    ,4E12.4)                                 AGIR2890
 110 FORMAT(37H0 ITERATION TERMINATING, ERROR WITHIN,F9.6,8H  BOUND.)   AGIR2900
 111 FORMAT(1X,7E12.4)                                                  AGIR2910
1000 FORMAT(1H1,26X,20A4,13X,A10/1H0,10X,15HSTARTING VALUES,5X,         AGIR2920
     -      6HMACH =,F6.3,5X,7HALPHA =,F7.2,5X,7HPARAM =,F10.4,5X,     AGIR2930
     -      4HCG =,F6.3)                                               AGIR2940
1001 FORMAT(15X,7A12)                                                   AGIR2950
2001 FORMAT(40H0ITERATION TERMINATING, MAXIMUM ERROR OF,E10.2,          AGIR2960
     -      9H EXCEEDED/27H0INPUT TIME HISTORY FOLLOWS)                AGIR2970
2003 FORMAT(45H0CONFIDENCE LEVELS FOR NEXT TO LAST ITERATION/           AGIR2980
     -      5X,13H(DIMENSIONAL)/)                                      AGIR2990
2006 FORMAT(22H0     (NON-DIMENSIONAL))                                AGIR3000
 600 RETURN                                                             AGIR3010
     END                                                                AGIR3020
```

SUBROUTINE OUTPUT

Description: Subroutine OUTPUT provides the final output in several forms. The time histories are computed with the final derivative estimates and may be printed or written on a file for plotting. Final derivative estimates are also printed and, if requested, punched on cards. An error exit section to print the input time history is entered if PLTMAX or ERRMAX was exceeded.

Programing notes: Time history data for plotting are written on unit 8. The time histories are always computed to determine the final error sum, even if neither printout nor plots are requested. Most variable names are similar to those in subroutine AGIRL. ERRVEC contains the error sum from each iteration in AGIRL for the convergence summary.

Subroutine listing:

```
      SUBROUTINE OUTPUT(D2)                                                 OUTP   0
C                                                                           OUTP  10
C     COMPUTES FINAL TIME HISTORY, OUTPUT MODES AS SPECIFIED                OUTP  20
C                                                                           OUTP  30
      COMMON /ALLDIM/ MAX,MIX                                               OUTP  40
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                                 OUTP  50
      COMMON /COM/ NCASE,MZ,NPTS,NPTT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,       OUTP  60
     -     D1TOL,D1RLX,NAPR,WFAC,WMAFR,ERRSUM,LAST,RATIO                    OUTP  70
      COMMON /INFO/ HH,NOITER,MX,MXP1,MU,TEST,XAN,ZAX,IPQR,IXYZ,XT3,MZM,OUTP  80
     -     CORECT,BIASKN,PLTMAX,XT4,ERRVEC,PUNCHD,NEAT                      OUTP  90
      COMMON/HEADNG/ LABELS,TITLE,JULIAN                                    OUTP 100
      COMMON /DIMENS/ C,CC,E,WQS,THETN                                      OUTP 110
      COMMON /ROUTH/ PUNCH ,PARAM,MACH,ALP,CG,AC,BC                         OUTP 120
      DIMENSION A(5,4),B(5,8),AA(5,4),BB(5,8),AP(4,4),BP(4,8),D1(8,7),      OUTP 130
     -               XT4(3),NPTS(15),PHI1(5,4),APHI(5,4),DUM1(5,4),       OUTP 140
     -     XT1(7),XT2(7),D2(7),Z(7),BJI(4),TITLE(20),U1(8),U2(8),           OUTP 150
     -     ERRVEC(20),AAP(3,4),BBP(3,8),XT6(7),U(4),AC(5,4),BC(5,8)         OUTP 160
      LOGICAL TEST,CORECT,BIASKN,LATR,PUNCH,PLOTEM,E(24),PRINT,LAST,        OUTP 170
     -     OUTPT,PUNCHD,LONG                                                OUTP 180
      REAL CALIB(7),LABELS(15),MACH,C(3,3),CC(3,5),U3(4),Y(7),BIASD(4)      OUTP 190
     -     ,RI(5,4),RIA(5,4),RIB(5,8),EXTRA(4),EXTR1(4),XT3(4)              OUTP 200
      DATA PHI1L/4HPHI1/,APHIL/4HAPHI/,ALAT/4HLATR/,ALON/4HLONG/            OUTP 210
      RAD=57.2958                                                           OUTP 220
      DO 5 I=1,7                                                            OUTP 230
    5 CALIB(I)=1./RAD                                                       OUTP 240
      CALIB(5)=1.                                                           OUTP 250
      IF(LATR) GO TO 7                                                      OUTP 260
      CALIB(3)=1.                                                           OUTP 270
      CALIB(7)=1.                                                           OUTP 280
    7 IF(NOITER.EQ.0) GO TO 8                                               OUTP 290
      IF(ERRVEC(NOITER).GE.PLTMAX) GO TO 450                                OUTP 300
    8 PHI1(5,3)=PHI1L                                                       OUTP 310
      APHI(5,3)=APHIL                                                       OUTP 320
      OUTPT=PRINT.OR.PLOTEM                                                 OUTP 330
      REWIND 7                                                              OUTP 340
      MAX=5                                                                 OUTP 350
      WRITE(3,1000)TITLE,JULIAN                                             OUTP 360
      WRITE(3,2000)MACH,ALP,PARAM,CG                                        OUTP 370
      CALL DERIV(LONG)                                                      OUTP 380
      DO 10 I=1,MZM                                                         OUTP 390
   10 BIASD(I)=XT4(I)/CALIB(I+MX)                                           OUTP 400
      IF(BIASKN) WRITE(3,1003)(XT4(I),I=1,MZM)                              OUTP 410
C************************************** FINAL TIME HISTORY                  OUTP 420
      CALL AMULT(RI,A,RIA)                                                  OUTP 430
      CALL AMULT(RI,B,RIB)                                                  OUTP 440
      CALL AEAT(RIA,HH,PHI1,APHI,DUM1,BBP,NEAT)                             OUTP 450
      CALL AMULT(APHI,PI,DUM1)                                              OUTP 460
      CALL AMAKE(APHI,DUM1)                                                 OUTP 470
      IF (.NOT.TEST    ) GO TO 60                                           OUTP 480
      CALL ASPIT(APHI)                                                      OUTP 490
      CALL ASPIT(PHI1)                                                      OUTP 500
   60 DO 70 I=1,MZ                                                          OUTP 510
   70 D2(I)=0.                                                              OUTP 520
      DO 55 I=1,3                                                           OUTP 530
      DO 52 J=1,MX                                                          OUTP 540
   52 AAP(I,J)=RIA(I,J)*AP(I,J)                                             OUTP 550
      DO 55 J=1,MU                                                          OUTP 560
```

```
   55 BBP(I,J)=RIB(I,J)*BP(I,J)                                          OUTP 570
      IF(.NOT.CORECT) GO TO 69                                           OUTP 580
      DO 61 J=1,MX                                                       OUTP 590
      AAP(1,J)=AAP(1,J)+RIA(2,J)*XAN                                     OUTP 600
   61 AAP(IXYZ,J)=AAP(IXYZ,J)+ZAX*RIA(IPQR,J)                            OUTP 610
      DO 62 J=1,MU                                                       OUTP 620
      BBP(1,J)=BBP(1,J)+XAN*RIB(2,J)                                     OUTP 630
   62 BBP(IXYZ,J)=BBP(IXYZ,J)+ZAX*RIB(IPQR,J)                            OUTP 640
   69 CONTINUE                                                           OUTP 650
      ERRSUM=0.                                                          OUTP 660
C************************************* CASE LOOP                        OUTP 670
      DO 200 LM=1,NCASE                                                  OUTP 680
      NNM1=NPTS(LM)-1                                                    OUTP 690
      READ (7) IT,XT6,U2,EXTRA                                           OUTP 700
      READ (7) IT1,XT2,U1,EXTR1                                          OUTP 710
      DO 75 I=1,MX                                                       OUTP 720
      Y(I)=XT6(I)+XT3(I)                                                 OUTP 730
   75 XT1(I)=XT2(I)+XT3(I)                                               OUTP 740
      IF(.NOT.OUTPT) GO TO 95                                            OUTP 750
      DO 76 I=MXP1,MZ                                                    OUTP 760
      Y(I)=XT6(I)                                                        OUTP 770
   76 XT1(I)=XT2(I)                                                      OUTP 780
      DO 80 I=1,MZ                                                       OUTP 790
      Z(I) = XT6(I)/CALIB(I)                                             OUTP 800
      Y(I)=Y(I)/CALIB(I)                                                 OUTP 810
      XT2(I)=XT2(I)/CALIB(I)                                             OUTP 820
   80 XT6(I) = XT1(I)/CALIB(I)                                           OUTP 830
      DO 91 I=1,4                                                        OUTP 840
      U3(I)=U2(I)*RAD                                                    OUTP 850
   91 U(I)=U1(I)*RAD                                                     OUTP 860
      IF(.NOT.PRINT) GO TO 93                                            OUTP 870
      WRITE(3,1000)TITLE,JULIAN                                          OUTP 880
      IF(LM.EQ.1) WRITE(3,2005)                                          OUTP 890
      WRITE(3,2004)(LABELS(I),I=1,MZ)                                    OUTP 900
      WRITE(3,113)IT,(Z(I),I=1,MZ)                                       OUTP 910
      WRITE(3,113)IT1,(XT6(I),I=1,MZ)                                    OUTP 920
      LINE=2                                                             OUTP 930
      IF(.NOT.PLOTEM) GO TO 95                                           OUTP 940
   93 WRITE (8) Y,Z,U3,EXTRA                                             OUTP 950
      WRITE (8) XT6,XT2,U,EXTR1                                          OUTP 960
C************************************* TIME LOOP                        OUTP 970
   95 DO 200 IP=2,NNM1                                                   OUTP 980
      READ (7) IT,Z,U2,EXTRA                                             OUTP 990
      DO 110 I=1,MX                                                      OUTP1000
      BJI(I)=0.                                                          OUTP1010
      XT2(I)=0.                                                          OUTP1020
      DO 110 J=1,MU                                                      OUTP1030
  110 BJI(I)=BJI(I)+B(I,J)*(U2(J)+U1(J))*.5                              OUTP1040
      DO 120 J=1,MX                                                      OUTP1050
      DO 120 K=1,MX                                                      OUTP1060
  120 XT2(J) = XT2(J) + BJI(K)*APHI(J,K)  +  XT1(K)*PHI1(J,K)            OUTP1070
      DO 140 L=MXP1,MZ                                                   OUTP1080
      LMMX=L-MX                                                          OUTP1090
      XT2(L)=XT4(LMMX)                                                   OUTP1100
      DO 130 K=1,MU                                                      OUTP1110
  130 XT2(L) = XT2(L) + U2(K)*BBP(LMMX,K)                                OUTP1120
      DO 140 K=1,MX                                                      OUTP1130
```

```
  140 XT2(L) = XT2(L) + XT2(K)*AAP(LMMX,K)                                  OUTP1140
      DO 150 J=1,MZ                                                        OUTP1150
      XT1(J) = XT2(J)                                                      OUTP1160
  150 D2(J) = D2(J) + (Z(J)-XT2(J))**2                                     OUTP1170
      IF(.NOT.OUTPT) GO TO 195                                             OUTP1180
      DO 170 I=1,MZ                                                        OUTP1190
      Z(I)=Z(I)/CALIB(I)                                                   OUTP1200
  170 Y(I)=XT2(I)/CALIB(I)                                                 OUTP1210
      DO 191 I=1,4                                                         OUTP1220
  191 U(I)=U2(I)*RAD                                                       OUTP1230
      IF(PLOTEM) WRITE (8) Y,Z,U,EXTRA                                     OUTP1240
      IF(.NOT.PRINT) GO TO 195                                             OUTP1250
      IF(LINE .LT. 50) GO TO 190                                           OUTP1260
      LINE = 0                                                             OUTP1270
      WRITE(3,1000)TITLE,JULIAN                                            OUTP1280
      WRITE(3,2004)(LABELS(I),I=1,MZ)                                      OUTP1290
  190 LINE = LINE+1                                                        OUTP1300
      WRITE(3,113)IT,(Y(I),I=1,MZ)                                         OUTP1310
  195 DO 200 K=1,MU                                                        OUTP1320
      U1(K) = U2(K)                                                        OUTP1330
  200 CONTINUE                                                             OUTP1340
C************************************* END LOOPS                           OUTP1350
      WRITE(3,2002)                                                        OUTP1360
      CALL ASPIT(A)                                                        OUTP1370
      CALL ASPIT(B)                                                        OUTP1380
C************************************* PUNCHED OUPUT AS DESIRED            OUTP1390
      IF(.NOT.PUNCHD) GO TO 300                                            OUTP1400
      CALL PLOP(A)                                                         OUTP1410
      CALL PLOP(B)                                                         OUTP1420
  300 IF(.NOT.PUNCH) GO TO 400                                             OUTP1430
      DO 320 I=1,3                                                         OUTP1440
      DO 310 J=1,3                                                         OUTP1450
  310 A (I,J)=A (I,J)*C(I,J)                                               OUTP1460
      DO 320 J=1,5                                                         OUTP1470
  320 B (I,J)=B (I,J)*CC(I,J)                                              OUTP1480
      A(5,1)=3.                                                            OUTP1490
      B(5,1)=3.                                                            OUTP1500
      TYPE=ALAT                                                            OUTP1510
      IF(.NOT.LONG) GO TO 330                                              OUTP1520
      TYPE=ALON                                                            OUTP1530
C     DETRIM AND CZ (GOOD ONLY FOR 2 DEGREE OF FREEDOM WITH NO EXTRA       OUTP1540
C                    CONTROLS.)                                            OUTP1550
      B(2,5)=-(A(2,1)*ALP+B(2,5))/B(2,1)                                   OUTP1560
      B(1,5)=B(1,5)+A(1,1)*ALP+B(1,1)*B(2,5)-WQS                           OUTP1570
  330 WRITE(2,2001)TYPE,(TITLE(I),I=1,9),MACH,ALP,PARAM,CG                 OUTP1580
      CALL PLOP(A )                                                        OUTP1590
      CALL PLOP(B )                                                        OUTP1600
      CALL PLOP(AC)                                                        OUTP1610
      CALL PLOP(BC)                                                        OUTP1620
  400 IF(.NOT.BIASKN) GO TO 209                                            OUTP1630
      WRITE(3,1004)(LABELS(I),I=MXP1,MZ)                                   OUTP1640
      WRITE(3,1003) (BIASD(I),I=1,MZM)                                     OUTP1650
  209 DO 210 I=1,MZ                                                        OUTP1660
      XT1(I)=D2(I)/FLOAT(NPTT)                                             OUTP1670
      D2(I)=XT1(I)*D1(I,I)                                                 OUTP1680
  210 ERRSUM = ERRSUM + D2(I)                                              OUTP1690
      WRITE(3,1001)ERRSUM                                                  OUTP1700
```

```
      WRITE(3,100)                                                      OUTP1710
      WRITE(3,105)(XT1(I),I=1,MZ)                                       OUTP1720
      WRITE(3,106)                                                      OUTP1730
      WRITE(3,105)(D2(I),I=1,MZ)                                        OUTP1740
      IF(NOITER.NE.0) WRITE(3,108)(ERRVEC(I),I=1,NOITER),ERRSUM         OUTP1750
      IF(ERRSUM.LT.PLTMAX .OR. .NOT.PLOTEM) RETURN                      OUTP1760
  450 WRITE(3,1002)PLTMAX                                               OUTP1770
      WRITE(3,1000)TITLE,JULIAN                                         OUTP1780
      WRITE(3,2004)(LABELS(I),I=1,MZ)                                   OUTP1790
      PLOTEM=.FALSE.                                                    OUTP1800
      REWIND 7                                                          OUTP1810
      DO 500 I=1,NPTT                                                   OUTP1820
      READ (7) IT,Z,U,EXTRA                                             OUTP1830
      DO 460 J=1,7                                                      OUTP1840
  460 Z(J)=Z(J)/CALIB(J)                                                OUTP1850
      DO 470 J=1,4                                                      OUTP1860
  470 U(J)=U(J)*RAD                                                     OUTP1870
  500 WRITE(3,113)IT,Z,U                                                OUYP1880
  100 FORMAT(7H ERRORS)                                                 OUTP1890
  105 FORMAT (1X,11E12.4)                                               OUTP1900
  106 FORMAT(16H WEIGHTED ERRORS)                                       OUTP1910
  108 FORMAT(1H0,62X,6HERRORS/(1X,13F10.2))                             OUTP1920
  113 FORMAT(2X,I12,11F10.4)                                            OUTP1930
 1000 FORMAT(1H1,26X,20A4,13X,A10/)                                     OUTP1940
 1001 FORMAT(90X,20HWEIGHTED ERROR SUM =,E12.4)                         OUTP1950
 1002 FORMAT(55H0DATA WILL NOT BE PLOTTED BECAUSE THE ERROR SUM EXCEEDS,OUTP1960
     -      24H THE MAXIMUM PERMISSIBLE,E10.2/                         OUTP1970
     -      27H0INPUT TIME HISTORY FOLLOWS)                            OUTP1980
 1003 FORMAT(15H VARIABLE BIAS ,4E12.4)                                 OUTP1990
 1004 FORMAT(8H0DEGREES,7X,4A12)                                        OUTP2000
 2000 FORMAT(1H0,10X,12HFINAL VALUES,5X,6HMACH =,F6.3,5X,7HALPHA =,F7.2,OUTP2010
     -      5X,7HPARAM =,F10.4,5X,4HCG =,F6.3)                         OUTP2020
 2001 FORMAT(A4,1X,8A4,A3,4F10.3)                                       OUTP2030
 2002 FORMAT(27H FINAL DIMENSIONAL MATRICES)                            OUTP2040
 2004 FORMAT(1H0,5X,4HTIME,10X,7A10/)                                   OUTP2050
 2005 FORMAT(20H OUTPUT TIME HISTORY)                                   OUTP2060
      RETURN                                                            OUTP2070
      END                                                               OUTP2080
```

SUBROUTINE THPLOT

Description: Subroutine THPLOT plots measured and computed time histories of observations and measured time histories of controls and extra signals.

Programing notes: The comment cards show how to decrease the run time in some instances at the cost of some storage. At present, two time histories at a time are read from the disk and plotted. Dimensions may be increased as indicated to permit more than two to be handled simultaneously, resulting in fewer disk accesses. With reasonably efficient disk units, the saving is not a significant portion of the program execution time. The limitation of 1000 points per maneuver arises from the dimensioning of X,XX,XXX and TIME as 1002. (The extra two locations are used for scaling information.) Program size may be reduced or the permissible maneuver length increased by changing this value. The special treatment of the title (plotting groups of four characters in a DO loop instead of using only one call to SYMBOL) is needed for compatibility with machines that use different word lengths.

Subroutine listing:

```
      SUBROUTINE THPLOT(FIRST)                                          THPL   0
C                                                                       THPL  10
C     PLOTS TIME HISTORIES                                              THPL  20
C                                                                       THPL  30
      COMMON /BUF/ BUFFER,YO,THGT                                       THPL  40
      COMMON /COM/ NCASE,MZ,NPTS,NPTT,SPS,PRINT,LONG,LATR,PLOTEM,ND1,   THPL  50
     -     D1TOL,D1RLX,NAPR,WFAC,WMAPR,ERRSUM,LAST,RATIO                THPL  60
      COMMON /TOPLOT/ XMAX,XMIN,DCMAX,DCMIN,TIMESC,NC                   THPL  70
      COMMON /HEADNG/ LABELS,TITLE,JULIAN                               THPL  80
      COMMON /LINCOM/ HGT                                               THPL  90
      DIMENSION DCMAX(8),DCMIN(8),XMAX(7),XMIN(7),NPTS(15),TITLE(20),   THPL 100
     -     BUFFER(1024),TIME(1002),XXX(1002,2),X(1002,1),XX(1002,1),    THPL 110
     -     Z(7),DC(8),ZZ(7),LABELS(15),MBCD(30)                         THPL 120
      LOGICAL LONG,FIRST,LAST                                           THPL 130
      EQUIVALENCE (X(1,1),XXX(1,1)),(XX(1,1),XXX(1,2))                  THPL 140
      DATA          MBCD/3HDEG,3HD/S,3HF/S,3HDEG,3HG'S,4HD/S2,3HG'S,    THPL 150
     -     5*3HDEG,2HFT,1H ,3HPSF,3HDEG,2*3HD/S,3HDEG,3HG'S,2*4HD/S2,   THPL 160
     -     5*3HDEG,4HFT/S,1H ,3HPSF/                                    THPL 170
      NCH=1                                                             THPL 180
C****************** FOR A DIRECT DECREASE IN RUN TIME AT THE COST OF    THPL 190
C                    STORAGE, NCH MAY BE INCREASED (UP TO 7). THEN THE  THPL 200
C                     FOLLOWING DIMENSIONS AND EQUIVALENCE MUST BE      THPL 210
C                    CHANGED.                                           THPL 220
C                    DIMENSION X(1002,NCH),XX(1002,NCH),XXX(1002,2*NCH) THPL 230
C****************** EQUIVALENCE (XX(1,1),XXX(1,NCH+1))                 THPL 240
      NBUF=1024                                                         THPL 250
      TIMSC2=TIMESC*2.                                                  THPL 260
      XO=5.                                                             THPL 270
      HGT=.01                                                           THPL 280
      NIP=0                                                             THPL 290
      TSI=SPS*TIMSC2                                                    THPL 300
      ITHIN=-MAX1(TSI/20.,1.)                                           THPL 310
      REWIND 8                                                          THPL 320
      IF(.NOT.FIRST) GO TO 10                                           THPL 330
      CALL PLOTS(BUFFER,NBUF,13)                                        THPL 340
      CALL FACTOR(RATIO)                                                THPL 350
      YO=12.                                                            THPL 360
      IF(RATIO.EQ.1.) YO=9.5                                            THPL 370
      THGT=.12/RATIO                                                    THPL 380
C****************** LABELS AND TITLES                                   THPL 390
   10 Y75=YO+.375                                                       THPL 400
      IF (LONG) GO TO 50                                                THPL 410
      DO 20 I=1,15                                                      THPL 420
   20 MBCD(I)=MBCD(I+15)                                                THPL 430
   50 DO 200 I=1,NCASE                                                  THPL 440
      CALL PLOT(XO,0.,-3)                                               THPL 450
      CALL SYMBOL(-1.5,YO,THGT,TITLE(1),270.,4)                         THPL 460
      DO 55 J=2,20                                                      THPL 470
   55 CALL SYMBOL(-1.5,999.,THGT,TITLE(J),270.,4)                       THPL 480
      IF(NCASE.EQ.1) GO TO 57                                           THPL 490
      CALL SYMBOL(-2.,YO,THGT,8HMANEUVER,270.,8)                        THPL 500
      YO15=YO-1.5                                                       THPL 510
      CALL NUMBER(-2.,YO15,THGT,FLOAT(I),270.,-1)                       THPL 520
   57 CALL PLTDAT(-2.5,YO)                                              THPL 530
C****************** FORM TIME VECTOR AND PLOT TIME AXIS                 THPL 540
      NOPTS=NPTS(I)                                                     THPL 550
      NP1=NOPTS+1                                                       THPL 560
```

```
      NP2=NOPTS+2                                                       THPL 570
      DO 60 J=1,NOPTS                                                   THPL 580
   60 TIME(J)=J                                                         THPL 590
      TIME(NP1)=YO*TSI+1.                                               THPL 600
      TIME(NP2)=-TSI                                                    THPL 610
      TLN=FLOAT(NOPTS)/TSI                                              THPL 620
      CALL AXIS(0.,YC,4HTIME,-4,TLN,270.,0.,TIMSC2)                     THPL 630
      XORG=.5                                                           THPL 640
C ****************** PLOT STATE TIME HISTORIES                          THPL 650
      ICHAN0=0                                                          THPL 660
      NCHAN=NCH                                                         THPL 670
      DO 120 K=1,7                                                      THPL 680
      IF(K .EQ. 1) GO TO 90                                             THPL 690
      ICHAN0=ICHAN0+NCHAN                                               THPL 700
      IF(ICHAN0+NCHAN .LE.MZ) GO TO 70                                  THPL 710
      IF(ICHAN0 .GE.MZ) GO TO 125                                       THPL 720
      NCHAN=MZ-ICHAN0                                                   THPL 730
   70 REWIND 8                                                          THPL 740
      IF(I .EQ. 1) GO TO 90                                             THPL 750
      DO 80 J=1,NIP                                                     THPL 760
   80 READ (8)                                                          THPL 770
   90 DO 100 J=1,NOPTS                                                  THPL 780
      READ (8) ZZ,Z,DC                                                  THPL 790
      DO 100 L=1,NCHAN                                                  THPL 800
      X(J,L)=Z(L+ICHAN0)                                                THPL 810
  100 XX(J,L)=ZZ(L+ICHAN0)                                              THPL 820
      DO 110 L=1,NCHAN                                                  THPL 830
      ICHAN=ICHAN0+L                                                    THPL 840
      SCAL =(XMAX(ICHAN)-XMIN(ICHAN))*.5                                THPL 850
      XMN=XMIN(ICHAN)                                                   THPL 860
      IF(SCAL.NE.0.) GO TO 105                                          THPL 870
      CALL SCALES(X(1,L),2.,NOPTS,.FALSE.)                              THPL 880
      CALL SCALES(XX(1,L),2.,NOPTS,.FALSE.)                             THPL 890
      SCAL=AMAX1(XX(NP2,L),X(NP2,L))                                    THPL 900
      IF(SCAL.EQ.-999.) GO TO 110                                       THPL 910
      XMN=X(NP1,L)                                                      THPL 920
      IF(XX(NP2,L).GT.X(NP2,L)) XMN=XX(NP1,L)                           THPL 930
  105 CALL SYMBOL(XORG+1.,Y75,.125,LABELS(ICHAN),0.,4)                  THPL 940
      CALL AXIS(XORG,YO,MBCD(ICHAN),4,2.,0.,XMN,SCAL)                   THPL 950
      X(NP1,L)=XMN-XORG*SCAL                                            THPL 960
      XX(NP1,L)=X(NP1,L)                                                THPL 970
        X(NP2,L)=SCAL                                                   THPL 980
      XX(NP2,L)=SCAL                                                    THPL 990
      CALL LINES(X(1,L),TIME,NOPTS,1, 0,1)                              THPL1000
      CALL LINES(XX(1,L),TIME,NOPTS,ITHIN,-2,75)                        THPL1010
      XORG=XORG+2.5                                                     THPL1020
  110 CONTINUE                                                          THPL1030
  120 CONTINUE                                                          THPL1040
C ******************  PLOT CONTROL TIME HISTORIES                       THPL1050
  125 NCH2=NCHAN*2                                                      THPL1060
      IF(NCH2.GT.NC) NCH2=NC                                            THPL1070
      ICHAN0=-NCH2                                                      THPL1080
      DO 160 K=1,4                                                      THPL1090
      ICHAN0=ICHAN0+NCH2                                                THPL1100
      IF(ICHAN0.GE.NC) GO TO 170                                        THPL1110
      IF(ICHAN0+NCH2.GT.NC) NCH2=NC-ICHAN0                              THPL1120
      REWIND 8                                                          THPL1130
```

```
      IF(I .EQ. 1) GO TO 140                                          THPL1140
      DO 130 J=1,NIP                                                  THPL1150
  130 READ (8)                                                        THPL1160
  140 DO 150 J=1,NOPTS                                                THPL1170
      READ (8) ZZ,Z,DC                                                THPL1180
      DO 150 L=1,NCH2                                                 THPL1190
  150 XXX(J,L)=DC(L+ICHAN0)                                           THPL1200
      DO 160 L=1,NCH2                                                 THPL1210
      J=L+ICHAN0                                                      THPL1220
      M=J+7                                                           THPL1230
      SCAL =(DCMAX(J)-DCMIN(J))*.5                                    THPL1240
      DCMN=DCMIN(J)                                                   THPL1250
      IF(SCAL.NE.0.) GO TO 155                                        THPL1260
      CALL SCALES(XXX(1,L),2.,NOPTS,.TRUE.)                           THPL1270
      IF(XXX(NP2,L).EQ.-999.) GO TO 160                               THPL1280
      IF(XXX(NP2,L).GE..4 .OR. J.EQ.7) GO TO 153                      THPL1290
      XXX(NP2,L)=10.                                                  THPL1300
      XXX(NP1,L)=-10.                                                 THPL1310
  153 SCAL=XXX(NP2,L)                                                 THPL1320
      DCMN=XXX(NP1,L)                                                 THPL1330
  155 CALL SYMBOL (XCRG+1.,Y75,.125,LABELS(M),0,4)                    THPL1340
      CALL AXIS(XORG,YO,MBCD(M),4,2.,0.,DCMN,SCAL)                    THPL1350
      XXX(NP1,L)=DCMN-XORG*SCAL                                       THPL1360
      XXX(NP2,L)=SCAL                                                 THPL1370
      CALL LINES(XXX(1,L),TIME,NOPTS,1, 0,1)                          THPL1380
      XORG=XORG+2.5                                                   THPL1390
  160 CONTINUE                                                        THPL1400
  170 NIP=NIP+NOPTS                                                   THPL1410
      XO=XORG+5.                                                      THPL1420
  200 CONTINUE                                                        THPL1430
      CALL PLOT(XO,0.,-3)                                             THPL1440
      RETURN                                                          THPL1450
      END                                                             THPL1460
```

SUBROUTINE APRPLT

Description: Subroutine APRPLT plots the variation of the derivatives with *a priori* weighting. It may be used when the *a priori* variation option is active. The information to be plotted is in the matrix STORE.

Subroutine listing:

```
      SUBROUTINE APRPLT(STORE,AA,BB,NAPR,WHOLD,WFAC,LONG,FIRST,LAST,     APRP   0
     -     RATIO)                                                        APRP  10
C                                                                        APRP  20
C     PLOTS DERIVATIVES FOR APRIORI VARIATION                            APRP  30
C                                                                        APRP  40
      COMMON /BUF/ BUFFER,YO,THGT                                        APRP  50
      COMMON/HEADNG/ LABELS,TITLE,JULIAN                                 APRP  60
      COMMON /LINCOM/ HGT                                                APRP  70
      LOGICAL AA(5,4),BB(5,8),LONG,FIRST,LAST                            APRP  80
      REAL STORE(14,27),ALABLO(3,3),ALABLA(3,3),BLABLO(3,4),BLABLA(3,4),APRP  90
     -     BUFFER(1024),WMAPR(14),LABELS(15),TITLE(20)                   APRP 100
      DATA ALABLA/2HYB,2HLB,2HNB,2HYP,2HLP,2HNP,2HYR,2HLR,2HNR/,BLABLA/ APRP 110
     -     3HYDA,3HLDA,3HNDA,3HYDR,3HLDR,3HNDR,4HYDC1,4HLDC1,4HNDC1,     APRP 120
     -     4HYDC2,4HLDC2,4HNDC2/,ALAELO/2HZA,2HMA,2HXA,2HZQ,2HMQ,2HXQ,   APRP 130
     -     2HZU,2HMU,2HXU/,BLABLO/3HZDE,3HMDE,3HXDE,3HZDC,3HMDC,3HXDC,   APRP 140
     -     4HZDC1,4HMDC1,4HXDC1,4HZDC2,4HMDC2,4HXDC2/                    APRP 150
      HGT=.07                                                            APRP 160
      NBUF=1024                                                          APRP 170
      JK=0                                                               APRP 180
      NPT=NAPR+1                                                         APRP 190
      NPT1=NPT+1                                                         APRP 200
      NPT2=NPT+2                                                         APRP 210
      IF(.NOT.FIRST) GO TO 5                                             APRP 220
      CALL PLOTS(BUFFER,NBUF,13)                                         APRP 230
      CALL FACTOR(RATIO)                                                 APRP 240
      YO=12.                                                             APRP 250
      IF(RATIO.EQ.1.) YO=9.5                                             APRP 260
      THGT=.12/RATIO                                                     APRP 270
    5 CALL PLOT(0.,YO,-3)                                                APRP 280
      CALL SYMBOL(-1.5,0.,THGT,TITLE(1),270.,4)                          APRP 290
      DO 7 J=2,20                                                        APRP 300
    7 CALL SYMBOL(-1.5,999.,THGT,TITLE(J),270.,4)                        APRP 310
      CALL PLTDAT(-4.,.5)                                                APRP 320
      WMAPR(1)=-1.                                                       APRP 330
      DO 10 I=1,NAPR                                                     APRP 340
   10 WMAPR(I+1)=WMAPR(I)-1.                                             APRP 350
      WMAPR(NPT1)=0.                                                     APRP 360
      WMAPR(NPT2)=1.                                                     APRP 370
      CALL NUMBER(-.5,-1.,.1,0.,270.,0)                                  APRP 380
      Y=-.75                                                             APRP 390
      IF(NAPR.LT.2) GO TO 220                                            APRP 400
      NAPR1=NAPR-1                                                       APRP 410
      W=WHOLD                                                            APRP 420
      DO 210 I=1,NAPR1                                                   APRP 430
      Y=Y-1.                                                             APRP 440
      CALL NUMBER(-.5,Y,.1,W,270.,3)                                     APRP 450
  210 W=W*WFAC                                                           APRP 460
  220 CONTINUE                                                           APRP 470
      CALL SYMBOL(-1.,Y/2.,.125,5HWMAPR,270.,5)                          APRP 480
      DO 200 I=1,3                                                       APRP 490
      DO 100 J=1,4                                                       APRP 500
      IF(BB(I,J)) GO TO 100                                              APRP 510
      JK=JK+1                                                            APRP 520
      CALL SCALES(STORE(1,JK),3.,NPT,.FALSE.)                            APRP 530
      DER=BLABLA(I,J)                                                    APRP 540
      IF(LONG) DER=BLABLO(I,J)                                           APRP 550
      CALL AXIS(0.,0.,DER, 4,3.,0.,STORE(NPT1,JK),STORE(NPT2,JK))        APRP 560
```

```
      CALL LINES(STORE(1,JK),WMAPR,NPT,1,1,1)                          APRP 570
      CALL PLOT(3.5,0.,-3)                                             APRP 580
  100 CONTINUE                                                         APRP 590
      DO 200 J=1,3                                                     APRP 600
      IF(AA(I,J)) GO TO 200                                            APRP 610
      JK=JK+1                                                          APRP 620
      CALL SCALES(STORE(1,JK),3.,NPT,.FALSE.)                          APRP 630
      DER=ALABLA(I,J)                                                  APRP 640
      IF(LONG) DER=ALABLO(I,J)                                         APRP 650
      CALL AXIS(0.,0.,DER, 4,3.,0.,STORE(NPT1,JK),STORE(NPT2,JK))      APRP 660
      CALL LINES(STORE(1,JK),WMAPR,NPT,1,1,1)                          APRP 670
      CALL PLOT(3.5,0.,-3)                                             APRP 680
  200 CONTINUE                                                         APRP 690
      WMAPR(NPT)=WMAPR(NPT1)                                           APRP 700
      WMAPR(NPT1)=WMAPR(NPT2)                                          APRP 710
      NPT2=NPT1                                                        APRP 720
      NPT1=NPT                                                         APRP 730
      NPT=NAPR                                                         APRP 740
      JK=JK+1                                                          APRP 750
      CALL SCALES(STORE(1,JK),3.,NPT,.FALSE.)                          APRP 760
      CALL AXIS(0.,0.,5HERROR,5,3.,0.,STORE(NPT1,JK),STORE(NPT2,JK))   APRP 770
      CALL LINES(STORE(1,JK),WMAPR,NPT,1,1,1)                          APRP 780
      CALL PLOT(5.,-12.,-3)                                            APRP 790
      RETURN                                                           APRP 800
      END                                                              APRP 810
```

SUBROUTINE MATLD

Description: Subroutine MATLD reads matrices from cards and identifies the matrices.

Programing notes: ABC contains the names of the matrices that may be read in. The program compares the name read with elements of ABC to determine which matrix is being input. The characters END are taken as an indication that this is the last case; any other word not identifiable as a valid matrix name signals the end of a case, implying more cases to follow. The values of ILD and ABC indicate the status of the matrix input.

Subroutine listing:

```
      SUBROUTINE MATLD(MATRX,ABC,ILD)                                   MATL   0
C                                                                       MATL  10
C     LOADS IN MATRICES - DETERMINES WHICH MATRIX IS BEING READ         MATL  20
C          PASSES STATUS INFORMATION BACK TO NREDIT                     MATL  30
C                                                                       MATL  40
      REAL MATRX(8,8),ABC(12)                                           MATL  50
      DATA END/3HEND/                                                   MATL  60
      ILD=-999                                                          MATL  70
      READ (1,1000) MATRX(8,3),II,JJ                                    MATL  80
      IF(MATRX(8,3).EQ.END) RETURN                                      MATL  90
      DO 10 I=1,12                                                      MATL 100
      IF(MATRX(8,3).NE.ABC(I)) GO TO 10                                 MATL 110
      ABC(I)=-99999.                                                    MATL 120
      ILD=I                                                             MATL 130
      GO TO 20                                                          MATL 140
   10 CONTINUE                                                          MATL 150
      ILD=999                                                           MATL 160
      RETURN                                                            MATL 170
   20 MATRX(8,1)=II                                                     MATL 180
      IF(JJ.NE.0) GO TO 25                                              MATL 190
C     DIAGONAL MATRIX                                                   MATL 200
      ILD=-ILD                                                          MATL 210
      MATRX(8,2)=MATRX(8,1)                                             MATL 220
      CALL AZOT(MATRX)                                                  MATL 230
      READ (1,1001) (MATRX(I,I),I=1,II)                                 MATL 240
      RETURN                                                            MATL 250
C     FULL MATRIX                                                       MATL 260
   25 MATRX(8,2)=JJ                                                     MATL 270
      DO 30 I=1,II                                                      MATL 280
   30 READ (1,1001) (MATRX(I,J),J=1,JJ)                                 MATL 290
      RETURN                                                            MATL 300
 1000 FORMAT(A4,4X,I2,I10)                                              MATL 310
 1001 FORMAT(8F10.4)                                                    MATL 320
      END                                                               MATL 330
```

SUBROUTINE MAK

Description: Subroutine MAK moves an input matrix from its temporary location in MATRX to its proper location in X. Subroutines MATLD and EDIT have determined what the proper location is for each matrix.

Subroutine listing:

```
      SUBROUTINE MAK(X,MATRX,MAX)                                    MAK    0
      REAL X(MAX,1),MATRX(8,8)                                       MAK   10
      CALL ASPIT(MATRX)                                              MAK   20
      X(MAX,3)=MATRX(8,3)                                            MAK   30
      X(MAX,1)=MATRX(8,1)                                            MAK   40
      X(MAX,2)=MATRX(8,2)                                            MAK   50
      II=MATRX(8,1)                                                  MAK   60
      JJ=MATRX(8,2)                                                  MAK   70
      DO 10 I=1,II                                                   MAK   80
      DO 10 J=1,JJ                                                   MAK   90
   10 X(I,J)=MATRX(I,J)                                              MAK  100
      RETURN                                                         MAK  110
      END                                                            MAK  120
```

SUBROUTINE DERIV

Description: Subroutine DERIV prints dimensional and nondimensional derivatives with labels. Arrays E and EE contain the characters " " or "*" to indicate, when printed, that a particular derivative is either varying or fixed, respectively.

Subroutine listing:

```
      SUBROUTINE DERIV(LONG)                                                DERI   0
C     PRINT DIMENSIONAL AND NON-DIMENSIONAL DERIVATIVES                     DERI  10
      COMMON /HEADNG/ LABELS,TITLE,JULIAN                                   DERI  20
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                                 DERI  30
      COMMON /DIMENS/ C,CC,E,EE,WQS,THETN                                   DERI  40
      DIMENSION C(3,3),CC(3,5),A(5,4),B(5,8),AP(4,4),AA(5,4),BB(5,8),       DERI  50
     -     BP(4,8),AN(3,4),BN(3,5),E(3,3),EE(3,5),D1(8,7)                   DERI  60
      REAL RI(5,4),LABELS(15),TITLE(20),LAB(3),LONLAB(3),LATLAB(3)          DERI  70
      LOGICAL LONG                                                          DERI  80
      DATA F/1H /,G/1HC/,LATLAB/1HY,1HL,1HN/,LONLAB/1HN,1HM,1HA/            DERI  90
      DO 10 I=1,3                                                           DERI 100
      LAB(I)=LATLAB(I)                                                      DERI 110
      DO 5 J=1,4                                                            DERI 120
    5 AN(I,J)=A(I,J)                                                        DERI 130
      DO 10 J=1,5                                                           DERI 140
   10 BN(I,J)=B(I,J)                                                        DERI 150
      IF(.NOT.LONG) GO TO 20                                                DERI 160
      DO 15 I=1,3                                                           DERI 170
      LAB(I)=LONLAB(I)                                                      DERI 180
      AN(1,I)=-AN(1,I)                                                      DERI 190
   15 AN(3,I)=-AN(3,I)                                                      DERI 200
      DO 17 I=1,5                                                           DERI 210
      BN(1,I)=-BN(1,I)                                                      DERI 220
   17 BN(3,I)=-BN(3,I)                                                      DERI 230
C     WRITE DIMENSIONAL DERIVATIVES                                         DERI 240
   20 WRITE(3,107)                                                          DERI 250
      WRITE(3,106)(LABELS(I),I=1,3),(LABELS(J),J=8,11)                      DERI 260
      WRITE(3,103)                                                          DERI 270
     -  (F,LAB(I),(AN(I,J),E(I,J),J=1,3),(BN(I,J),EE(I,J),J=1,5),I=1,3)     DERI 280
C     NON-DIMENSIONALIZE                                                    DERI 290
      DO 29 I = 1,3                                                         DERI 300
      DO 25 J=1,5                                                           DERI 310
   25 BN(I,J)=BN(I,J)*CC(I,J)                                               DERI 320
      DO 29 J=1,3                                                           DERI 330
   29 AN(I,J)=AN(I,J)*C(I,J)                                                DERI 340
      IF(.NOT.LONG) AN(1,2)=0.                                              DERI 350
      WRITE(3,101)                                                          DERI 360
      WRITE(3,106)(LABELS(I),I=1,3),(LABELS(J),J=8,11)                      DERI 370
      WRITE(3,103)                                                          DERI 380
     -  (G,LAB(I),(AN(I,J),E(I,J),J=1,3),(BN(I,J),EE(I,J),J=1,5),I=1,3)     DERI 390
      WRITE(3,108)                                                          DERI 400
  101 FORMAT(48HONON-DIMENSIONAL DERIVATIVES / DEG (ROTARY /RAD))           DERI 410
  103 FORMAT((1X,2A1,8(F14.6,A1,1X)))                                       DERI 420
  106 FORMAT(7(11X,A5),11X,7HDELTA-0)                                       DERI 430
  107 FORMAT(39H0DIMENSIONAL DERIVATIVES / SEC / SEC**2)                    DERI 440
  108 FORMAT(/40X,50H( * : INDICATES DERIVATIVE HELD FIXED DURING MATCH)    DERI 450
      RETURN                                                                DERI 460
      END                                                                   DERI 470
```

SUBROUTINE CRAMER

Description: Subroutine CRAMER computes confidence levels based on Cramèr-Rao bounds.

Programing notes: The comment cards trace the steps of the subroutine. Note the manipulation of the SUM matrix required to store the second gradient (Hessian) with the *a priori* terms included, while also using the second gradient without the *a priori* terms for the confidence level computation.

Subroutine listing:

```
      SUBROUTINE CRAMER(SUM,APRO,MU,MZ,ERRSUM)                          CRAM   0
C                                                                       CRAM  10
C     COMPUTES CRAMER RAO BOUNDS (CONFIDENCE LEVELS)                    CRAM  20
C                                                                       CRAM  30
      COMMON /ALLDIM/ MAX,MIX                                           CRAM  40
      COMMON /MATRIX/ A,B,AA,BB,AP,BP,D1,RI                             CRAM  50
      COMMON /ROUTH/ PUNCH ,PARAM,MACH,ALP,CG,AC,BC                     CRAM  60
      DIMENSION A(5,4),B(5,8),AA(5,4),BB(5,8),AP(4,4),BP(4,8),D1(8,7),  CRAM  70
     -     SUM(35,35),AC(5,4),BC(5,8),APRO(35),RI(5,4)                  CRAM  80
      LOGICAL AA,BB,PUNCH                                               CRAM  90
      DATA ACL/2HAC/,BCL/2HBC/                                          CRAM 100
      AC(5,1)=3.                                                        CRAM 110
      AC(5,2)=3.                                                        CRAM 120
      BC(5,1)=3.                                                        CRAM 130
      BC(5,2)=5.                                                        CRAM 140
      AC(5,3)=ACL                                                       CRAM 150
      BC(5,3)=BCL                                                       CRAM 160
      JKMM1=SUM(MAX,1)                                                  CRAM 170
      JK2=JKMM1-1                                                       CRAM 180
C******************** SUBTRACT OUT APRIORI CONTRIBUTION TO HESSIAN      CRAM 190
C******************** STORE COMPLETE HESSIAN TEMPORARILY IN             CRAM 200
C******************** APRIORI LOCATIONS (UPPER TRIANGULAR) SINCE LAST   CRAM 210
C******************** USE HAS BEEN MADE OF IT.                          CRAM 220
      TEMP=SUM(JKMM1,JKMM1)                                             CRAM 230
      SUM(JKMM1,JKMM1)=SUM(JKMM1,JKMM1)-APRO(JKMM1)                     CRAM 240
      APRO(JKMM1)=TEMP                                                  CRAM 250
      DO 10 I=1,JK2                                                     CRAM 260
      TEMP=SUM(I,I)                                                     CRAM 270
      SUM(I,I)=SUM(I,I)-APRO(I)                                         CRAM 280
      APRO(I)=TEMP                                                      CRAM 290
      IP1=I+1                                                           CRAM 300
      DO 10 J=IP1,JKMM1                                                 CRAM 310
      TEMP=SUM(J,I)                                                     CRAM 320
      SUM(J,I)=SUM(J,I)-SUM(I,J)                                        CRAM 330
   10 SUM(I,J)=TEMP                                                     CRAM 340
C******************** OBTAIN DIAGONAL ELEMENTS OF INVERSE               CRAM 350
      CALL DIAGIN(SUM)                                                  CRAM 360
C******************** COMPUTE BOUNDS                                    CRAM 370
      WTS = 0.0                                                         CRAM 380
      DO 30029 I = 1,MZ                                                 CRAM 390
      IF (D1(I,I).NE.0.0) WTS = WTS + 1.0                               CRAM 400
30029 CONTINUE                                                          CRAM 410
      COEFF = ERRSUM / WTS                                              CRAM 420
      L=0                                                               CRAM 430
      DO 71 I=1,3                                                       CRAM 440
      DO 60 J=1,MU                                                      CRAM 450
      BC(I,J)=0.                                                        CRAM 460
      IF(BB(I,J)        ) GO TO 60                                      CRAM 470
      L=L+1                                                             CRAM 480
      BC(I,J)=SQRT(ABS(SUM(L,L))*COEFF)                                 CRAM 490
   60 CONTINUE                                                          CRAM 500
      DO 70 J=1,3                                                       CRAM 510
      AC(I,J)=0.                                                        CRAM 520
      IF(AA(I,J)        ) GO TO 70                                      CRAM 530
      L=L+1                                                             CRAM 540
      AC(I,J)=SQRT(ABS(SUM(L,L))*COEFF)                                 CRAM 550
   70 CONTINUE                                                          CRAM 560
```

```
      IF(.NOT.AA(I,4) ) L=L+1                                          CRAM 570
   71 CONTINUE                                                         CRAM 580
C******************* RESTORE COMPLETE HESSIAN TO LOWER TRIANGULAR PART  CRAM 590
      SUM(JKMM1,JKMM1)=APRO(JKMM1)                                     CRAM 600
      DO 80 I=1,JK2                                                    CRAM 610
      SUM(I,I)=APRO(I)                                                 CRAM 620
      IP1=I+1                                                          CRAM 630
      DO 80 J=IP1,JKMM1                                                CRAM 640
   80 SUM(J,I)=SUM(I,J)                                                CRAM 650
      RETURN                                                           CRAM 660
      END                                                              CRAM 670
```

SUBROUTINE AEAT

Description: Subroutine AEAT computes $e^{A\Delta t}$ and $\int_0^{\Delta t} e^{A\tau} d\tau$ using the Taylor series expansion.

Programing notes: The computational method used when NEAT $\neq$ 0 is described in the NAMELIST option NEAT (item (26), p. 15). The two matrices desired are returned as PHI and APHI. A2 and A3 are temporary scratch storage.

Subroutine listing:

```
      SUBROUTINE AEAT(A,TT,PHI,APHI,A2,A3,NEAT)                    AEAT   0
      COMMON /ALLDIM/ MAX,MIX                                      AEAT  10
      DIMENSION A(1),PHI(1),A2(1),APHI(1),A3(1)                    AEAT  20
              MAX2 = MAX * 2                                       AEAT  30
      II=A(MAX)                                                    AEAT  40
      PHI(MAX)=A(MAX)                                              AEAT  50
      PHI(MAX2)=A(MAX)                                             AEAT  60
      T=TT/(2.**NEAT)                                              AEAT  70
      CALL AZOT(PHI)                                               AEAT  80
      CALL AMAKE(APHI,PHI)                                         AEAT  90
      MI=-MAX                                                      AEAT 110
      DO 1 I = 1,II                                                AEAT 120
      MI=MI+MAX                                                    AEAT 130
        PHI(MI + I) = 1.                                           AEAT 140
    1 CONTINUE                                                     AEAT 150
      CALL AMAKE(A2,PHI)                                           AEAT 160
      G = 1.0                                                      AEAT 170
      DO 2 I=1,10                                                  AEAT 180
      BB = I                                                       AEAT 190
      G = G*T/BB                                                   AEAT 200
      CALL AADD(1.,APHI,G,A2,APHI)                                 AEAT 210
      CALL AMULT(A,A2,A3)                                          AEAT 220
      CALL AMAKE(A2,A3)                                            AEAT 230
      CALL AADD(1.,PHI,G,A2,PHI)                                   AEAT 240
    2 CONTINUE                                                     AEAT 250
      IF(NEAT.EQ.0) RETURN                                         AEAT 260
      DO 20 I=1,NEAT                                               AEAT 270
      CALL AMAKE(A2,PHI)                                           AEAT 280
      CALL AMULT(A2,A2,PHI)                                        AEAT 290
      MI=-MAX                                                      AEAT 300
      DO 10 J=1,II                                                 AEAT 310
      MI=MI+MAX                                                    AEAT 320
   10 A2(MI+J)=A2(MI+J)+1.                                         AEAT 330
      CALL AMULT(A2,APHI,A3)                                       AEAT 340
      CALL AMAKE(APHI,A3)                                          AEAT 350
   20 CONTINUE                                                     AEAT 360
      RETURN                                                       AEAT 370
      END                                                          AEAT 380
```

SUBROUTINE AMULT

Description: Subroutine AMULT computes C = A*B. The quantity C cannot be the same matrix as either A or B.

Subroutine listing:

```
      SUBROUTINE AMULT(A,B,C)                                    AMUL   0
      COMMON /ALLOIM/ MAX,MIX                                    AMUL  10
      REAL A(1),B(1),C(1)                                        AMUL  20
      MAX2=MAX*2                                                 AMUL  30
      MIX2=MIX*2                                                 AMUL  40
      II=A(MAX)                                                  AMUL  50
      C(MAX)=A(MAX)                                              AMUL  60
      JJ=A(MAX2)                                                 AMUL  70
      KK=B(MIX2)                                                 AMUL  80
      C(MAX2)=B(MIX2)                                            AMUL  90
      JE=(JJ-1)*MAX                                              AMUL 100
      KE=(KK-1)*MAX                                              AMUL 110
      DO 20 I=1,II                                               AMUL 120
      KEND=KE+I                                                  AMUL 130
      JEND=JE+I                                                  AMUL 140
      L=1                                                        AMUL 150
      DO 20 K=I,KEND,MAX                                         AMUL 160
      C(K)=0.                                                    AMUL 170
      JB=L                                                       AMUL 180
      DO 10 J=I,JEND,MAX                                         AMUL 190
      C(K)=A(J)*B(JB)+C(K)                                       AMUL 200
   10 JB=JB+1                                                    AMUL 210
   20 L=L+MIX                                                    AMUL 220
      RETURN                                                     AMUL 230
      END                                                        AMUL 240
```

SUBROUTINE DMULT

Description: Subroutine DMULT multiplies XJI by a diagonal matrix D1.

Subroutine listing:

```
      SUBROUTINE DMULT(XJI,D1,XJID1,JKM,MZ)                    DMUL   0
      REAL XJI(35,8),XJID1(35,7),D1(8,7)                       DMUL  10
      DO 10 I=1,MZ                                             DMUL  20
      DO 10 J=1,JKM                                            DMUL  30
   10 XJID1(J,I)=XJI(J,I)*D1(I,I)                              DMUL  40
      RETURN                                                   DMUL  50
      END                                                      DMUL  60
```

SUBROUTINE SUMULT

Description: Subroutine SUMULT adds the term XJID1*XJI* to the SUM matrix. Only the lower triangular elements are accumulated because the result must always be symmetrical.

Subroutine listing:

```
      SUBROUTINE SUMULT(XJI,XJID1,SUM,JKM,MZ)                  SUMU   0
      REAL XJI(35,8),XJID1(35,7),SUM(35,35)                    SUMU  10
      DO 10 I=1,JKM                                            SUMU  20
      DO 10 J=1,I                                              SUMU  30
      DO 10 K=1,MZ                                             SUMU  40
   10 SUM(I,J)=SUM(I,J)+XJID1(I,K)*XJI(J,K)                    SUMU  50
      RETURN                                                   SUMU  60
      END                                                      SUMU  70
```

SUBROUTINE PLOP

Description: Subroutine PLOP punches a matrix on cards.

Subroutine listing:

```
      SUBROUTINE PLOF(X)                                          PLOP   0
      COMMON /ALLDIM/ MAX,MIX                                     PLOP  10
      DIMENSION X(1)                                              PLOP  20
  102 FORMAT (8F10.6)                                             PLOP  30
  103 FORMAT(A4,4X,I2,I10)                                        PLOP  40
      MAX2=MAX+MAX                                                PLOP  50
      MAX3=MAX2+MAX                                               PLOP  60
      II=X(MAX)                                                   PLOP  70
      JJ=X(MAX2)                                                  PLOP  80
      WRITE(2,103)X(MAX3),II,JJ                                   PLOP  90
      KE=(JJ-1)*MAX                                               PLOP 100
      DO 2 I=1,II                                                 PLOP 110
      KEND=I+KE                                                   PLOP 120
    2 WRITE(2,102)(X(K),K=I,KEND,MAX)                             PLOP 130
      RETURN                                                      PLOP 140
      END                                                         PLOP 150
```

SUBROUTINE ASPIT

Description: Subroutine ASPIT prints a matrix.

Subroutine listing:

```
      SUBROUTINE ASPIT(X)                                         ASPI   0
C WRITES OUT MATRICES                                             ASPI  10
      COMMON /ALLDIM/ MAX,MIX                                     ASPI  20
      DIMENSION X(1)                                              ASPI  30
  100 FORMAT(1X,A4,30X,I3,4H  BY,I3)                              ASPI  40
  101 FORMAT (12X,10E12.4)                                        ASPI  50
                MAX2 = MAX * 2                                    ASPI  60
      MAX3=MAX2+MAX                                               ASPI  70
      II=X(MAX)                                                   ASPI  80
      JJ=X(MAX2)                                                  ASPI  90
      WRITE(3,100)X(MAX3),II,JJ                                   ASPI 100
      KE=(JJ-1)*MAX                                               ASPI 110
      DO 1 I =1,II                                                ASPI 120
      KEND=I+KE                                                   ASPI 130
    1 WRITE(3,101)(X(K),K=I,KEND,MAX)                             ASPI 140
      RETURN                                                      ASPI 150
      END                                                         ASPI 160
```

SUBROUTINE AADD

Description: Subroutine AADD adds scalar multiples of two matrices. $Z = g*X + h*Y$ with $g = 1$.

Subroutine listing:

```
      SUBROUTINE AADD (G,X,H,Y,Z)                                    AADD   0
C     SPECIALIZED VERSION FOR NR ASSUMES G=1.                        AADD  10
      COMMON /ALLDIM/ MAX,MIX                                        AADD  20
      DIMENSION X(1),Y(1),Z(1)                                       AADD  30
               MAX2 = MAX * 2                                        AADD  40
                  II = X(MAX)                                        AADD  50
                 JJ = X(MAX2)                                        AADD  60
      JEND=(JJ-1)*MAX+1                                              AADD  70
      IIM1=II-1                                                      AADD  80
      DO 53 J=1,JEND,MAX                                             AADD  90
      KEND=J+IIM1                                                    AADD 100
      DO  53 K=J,KEND                                                AADD 110
   53 Z(K)=X(K)+H*Y(K)                                               AADD 120
      Z(MAX)=X(MAX)                                                  AADD 130
      Z(MAX2)=X(MAX2)                                                AADD 140
      RETURN                                                         AADD 150
      END                                                            AADD 160
```

SUBROUTINE AZOT

Description: Subroutine AZOT sets all elements of a matrix to 0.

Subroutine listing:

```
      SUBROUTINE AZOT(X)                                             AZOT   0
      COMMON /ALLDIM/ MAX,MIX                                        AZOT  10
      DIMENSION X(1)                                                 AZOT  20
               MAX2 = MAX * 2                                        AZOT  30
      IIM1=X(MAX)-1.                                                 AZOT  40
      JJM1=X(MAX2)-1.                                                AZOT  50
      LEND=JJM1*MAX+1                                                AZOT  60
      DO 1 L=1,LEND,MAX                                              AZOT  70
      KEND=L+IIM1                                                    AZOT  80
      DO 1 K=L,KEND                                                  AZOT  90
    1 X(K)=0.                                                        AZOT 100
      RETURN                                                         AZOT 110
      END                                                            AZOT 120
```

SUBROUTINE AMAKE

Description: Subroutine AMAKE moves the matrix Y into X.

Subroutine listing:

```
      SUBROUTINE AMAKE(X,Y)                                    AMAK   0
      COMMON /ALLDIM/ MAX,MIX                                  AMAK  10
      DIMENSION X(1),Y(1)                                      AMAK  20
                MAX2 = MAX * 2                                 AMAK  30
      IIM1=Y(MAX)-1.                                           AMAK  40
      JJM1=Y(MAX2)-1.                                          AMAK  50
      LEND=JJM1*MAX+1                                          AMAK  60
      DO 1 L=1,LEND,MAX                                        AMAK  70
      KEND=L+IIM1                                              AMAK  80
      DO 1 K=L,KEND                                            AMAK  90
    1 X(K)=Y(K)                                                AMAK 100
      X(MAX)=Y(MAX)                                            AMAK 110
      X(MAX2)=Y(MAX2)                                          AMAK 120
      RETURN                                                   AMAK 130
      END                                                      AMAK 140
```

SUBROUTINE INV

Description: Subroutine INV inverts a general matrix in place.

Programing notes: Gauss elimination is used here; there is no pivoting, since this subroutine will be called only for a well-conditioned, near-diagonal matrix (the R matrix). See reference 9 for a discussion of this method.

Subroutine listing:

```
      SUBROUTINE INV(A,MAX)                                    INV    0
C     INVERTS A GENERAL MATRIX IN PLACE                        INV   10
C     NO PIVOTING (DIAGONAL ELEMENTS MUST BE NON-ZERO)         INV   20
      DIMENSION A(MAX,1)                                       INV   30
      N=A(MAX,1)                                               INV   40
      DO 80 K=1,N                                              INV   50
      BIGA=A(K,K)                                              INV   60
      DO 50 I=1,N                                              INV   70
      IF(I.EQ.K) GO TO 50                                      INV   80
      A(I,K)=-A(I,K)/BIGA                                      INV   90
   50 CONTINUE                                                 INV  100
      DO 60 I=1,N                                              INV  110
      IF(I.EQ.K) GO TO 60                                      INV  120
      DO 55 J=1,N                                              INV  130
      IF (J.EQ.K) GO TO 55                                     INV  140
      A(I,J)=A(I,J)+A(I,K)*A(K,J)                              INV  150
   55 CONTINUE                                                 INV  160
   60 CONTINUE                                                 INV  170
      DO 70 J=1,N                                              INV  180
      IF(J.EQ.K) GO TO 70                                      INV  190
      A(K,J)=A(K,J)/BIGA                                       INV  200
   70 CONTINUE                                                 INV  210
   80 A(K,K)=1./BIGA                                           INV  220
      RETURN                                                   INV  230
      END                                                      INV  240
```

SUBROUTINE SOLVE

Description: Subroutine SOLVE solves the system of linear equations, $Ax = b$, where A is symmetrical. It uses Cholesky's matrix decomposition. (See programing notes for subroutine REDUCE.) Only the lower triangular and diagonal elements of A are used.

Programing notes: The b vector is assumed to be stored as the N + 1 column of A, where N is the dimension of the system.

Subroutine listing:

```
      SUBROUTINE SOLVE(A,X)                                                  SOLV   0
C                                                                            SOLV  10
C     SOLVES SYSTEM AX=B (A SYMMETRIC, B STORED IN N+1 COLUMN OF A)          SOLV  20
C                                                                            SOLV  30
      REAL A(35,1),X(35)                                                     SOLV  40
      CALL REDUCE (A)                                                        SOLV  50
      N=A(35,1)                                                              SOLV  60
      NM1=N-1                                                                SOLV  70
      NP1=N+1                                                                SOLV  80
C****************** MULTIPLY (L)*(B)                                        SOLV  90
      DO 70 I=2,N                                                            SOLV 100
      X(I)=A(I,NP1)                                                          SOLV 110
      IM1=I-1                                                                SOLV 120
      DO 70 J=1,IM1                                                          SOLV 130
   70 X(I)=X(I)+A(I,J)*A(J,NP1)                                              SOLV 140
C****************** MULTIPLY BY (DI)                                        SOLV 150
      A(1,NP1)=A(1,NP1)/A(1,1)                                               SOLV 160
      DO 80 I=2,N                                                            SOLV 170
   80 A(I,NP1)=X(I)/A(I,I)                                                   SOLV 180
C****************** MULTIPLY BY (L*) TO FORM (L*)*(DI)*(L)*(B)              SOLV 190
      DO 90 I=1,NM1                                                          SOLV 200
      X(I)=A(I,NP1)                                                          SOLV 210
      IP1=I+1                                                                SOLV 220
      DO 90 J=IP1,N                                                          SOLV 230
   90 X(I)=X(I)+A(J,I)*A(J,NP1)                                              SOLV 240
      X(N)=A(N,NP1)                                                          SOLV 250
      RETURN                                                                 SOLV 260
      END                                                                    SOLV 270
```

SUBROUTINE DIAGIN

Description: Subroutine DIAGIN obtains the diagonal elements of the inverse of a symmetric matrix. It uses Cholesky's decomposition of the matrix. (See subroutine REDUCE programing notes.)

Subroutine listing:

```
      SUBROUTINE DIAGIN(A)                                          DIAG   0
C                                                                   DIAG  10
C     FIND DIAGONAL ELEMENTS OF A INVERSE FOR SYMMETRIC A           DIAG  20
C                                                                   DIAG  30
      REAL A(35,1)                                                  DIAG  40
      CALL REDUCE (A)                                               DIAG  50
      N=A(35,1)                                                     DIAG  60
      NM1=N-1                                                       DIAG  70
      DO 90 I=1,NM1                                                 DIAG  80
      A(I,I)=1./A(I,I)                                              DIAG  90
      IP1=I+1                                                       DIAG 100
      DO 90 J=IP1,N                                                 DIAG 110
   90 A(I,I)=A(I,I)+A(J,I)**2/A(J,J)                                DIAG 120
      A(N,N)=1./A(N,N)                                              DIAG 130
      RETURN                                                        DIAG 140
      END                                                           DIAG 150
```

SUBROUTINE REDUCE

Description: Subroutine REDUCE factors a symmetric matrix A by Cholesky's matrix decomposition.

Programing notes: The matrix is factored into $L^{-1}DL^{-1*}$, where L is the lower diagonal with unity diagonal elements, and D is diagonal. The lower diagonal, L, is returned in the lower triangular locations of A, except for the diagonal locations, which contain D.

Subroutine listing:

```
      SUBROUTINE REDUCE(A)                                              REDU   0
C                                                                       REDU  10
C     REDUCES SYMMETRIC MATRIX A STORED IN LOWER TRIANGULAR LOCATIONS   REDU  20
C     TO THE FORM (LI)*(D)*(LI*) WHERE L IS A LOWER TRIANGULAR MATRIX   REDU  30
C          WITH UNITY DIAGONAL TERMS, D IS A DIAGONAL MATRIX,           REDU  40
C          I DENOTES INVERSE AND * TRANSPOSE                            REDU  50
C                                                                       REDU  60
      REAL A(35,1)                                                      REDU  70
      N=A(35,1)                                                         REDU  80
      NM1=N-1                                                           REDU  90
      DO 20 K=1,NM1                                                     REDU 100
      KP1=K+1                                                           REDU 110
      KM1=K-1                                                           REDU 120
      AKKI=1./A(K,K)                                                    REDU 130
      DO 20 I=KP1,N                                                     REDU 140
      AKKIK=A(I,K)*AKKI                                                 REDU 150
      DO 10 J=I,N                                                       REDU 160
   10 A(J,I)=A(J,I)-AKKIK*A(J,K)                                        REDU 170
      A(I,K)=-AKKIK                                                     REDU 180
      IF(KM1.EQ.0) GO TO 20                                             REDU 190
      DO 15 J=1,KM1                                                     REDU 200
   15 A(I,J)=A(I,J)-AKKIK*A(K,J)                                        REDU 210
   20 CONTINUE                                                          REDU 220
C****************** L IS NOW STORED IN LOWER TRIANGULAR PART OF A       REDU 230
C****************** EXCEPT FOR DIAGONAL, WHICH CONTAINS D               REDU 240
      RETURN                                                            REDU 250
      END                                                               REDU 260
```

SUBROUTINE SCALES

Description: Subroutine SCALES determines scales for plotting the vector X on an axis S inches long. If the formal parameter ZERO is true, 0 must be included in the scale.

Programing notes: The minimum value on the axis is returned in location X(N + 1), and the scale in units per inch is returned in location X(N + 2). The only scales permitted are 2, 4, and 10 units per inch times a multiple of 10. A -999. is returned to indicate that all values of X are the same.

Subroutine listing:

```
      SUBROUTINE SCALES(X,S,N,ZERO)                                  SCAL   0
      LOGICAL ZERO                                                   SCAL  10
      REAL X(1),FAC(3)                                               SCAL  20
      DATA FAC/2.,4.,10./                                            SCAL  30
      XMAX=X(1)                                                      SCAL  40
      XMIN=X(1)                                                      SCAL  50
      IF(.NOT.ZERO) GO TO 10                                         SCAL  60
      XMAX=0.                                                        SCAL  70
      XMIN=0.                                                        SCAL  80
   10 DO 20 I=1,N                                                    SCAL  90
      XMAX=AMAX1(XMAX,X(I))                                          SCAL 100
   20 XMIN=AMIN1(XMIN,X(I))                                          SCAL 110
      A=XMAX-XMIN                                                    SCAL 120
      IF(A.NE.0.) GO TO 30                                           SCAL 130
      SCALE=-999.                                                    SCAL 140
      GO TO 100                                                      SCAL 150
   30 B=A/S                                                          SCAL 160
      J=IFIX(ABS(ALOG10(B)))                                         SCAL 170
      IF(B.LT.1.) J=-J-1                                             SCAL 180
      FACT=10.**J                                                    SCAL 190
      B=B/FACT                                                       SCAL 200
      DO 50 I=1,3                                                    SCAL 210
      SCALE=FACT*FAC(I)                                              SCAL 220
      AMIN=XMIN-AMOD(XMIN,SCALE)                                     SCAL 230
      IF(AMIN.GT.XMIN) AMIN=AMIN-SCALE                               SCAL 240
      IF((XMAX-AMIN).LE.SCALE*S) GO TO 100                           SCAL 250
   50 CONTINUE                                                       SCAL 260
      SCALE=10.*FACT*FAC(1)                                          SCAL 270
      AMIN=XMIN-AMOD(XMIN,SCALE)                                     SCAL 280
      IF(AMIN.GT.XMIN) AMIN=AMIN-SCALE                               SCAL 290
  100 X(N+1)=AMIN                                                    SCAL 300
      X(N+2)=SCALE                                                   SCAL 310
      RETURN                                                         SCAL 320
      END                                                            SCAL 330
```

SUBROUTINE LINES

Description: Subroutine LINES plots solid or dashed lines or symbols of the X-axis versus the Y-axis.

Programing notes: The quantities X and Y are assumed to have scaling information in locations NPT + 1 and NPT + 2 as placed there by subroutine SCALES or other sources. Every ISKIP point of the data is used, and the sign of ISKIP determines whether the plot is to be made starting from the lower numbered locations in the arrays (positive sign) or the higher numbered locations (negative sign). If JSKIP = 0, a solid line is plotted; if positive, a solid line is plotted with symbols every JSKIPth point. If JSKIP = -1, only symbols are plotted. A dashed line may be plotted using JSKIP = -2. The quantity L indicates the symbol to be plotted if relevant, and HGT gives its height.

Subroutine listing:

```
      SUBROUTINE LINES(X,Y,NPT,ISKIP,JSKIP,L)                                  LINE   0
C     ISKIP=+ PLOT FORWARD,- BACKWARDS                                         LINE  10
C     JSKIP= 0 LINE ONLY,+ LINE AND SYMBOLS, - SYMBOLS ONLY OR DASHED LNLINE  20
      COMMON /LINCOM/ HGT                                                      LINE  30
      REAL X(1),Y(1)                                                           LINE  40
      LOGICAL SYMB                                                             LINE  50
      IF(ABS(HGT-.5).GE..5) HGT=.07                                            LINE  60
      XMIN=X(NPT+1)                                                            LINE  70
      YMIN=Y(NPT+1)                                                            LINE  80
      DX=X(NPT+2)                                                              LINE  90
      DY=Y(NPT+2)                                                              LINE 100
      IS=IABS(ISKIP)                                                           LINE 110
      N=(NPT-1)/IS+1                                                           LINE 120
      NA=1                                                                     LINE 130
      IF(ISKIP.LT.0) NA=IS*(N-1)+1                                             LINE 140
      JMOD=MAX0(IABS(JSKIP),1)*IS                                              LINE 150
      SYMB=.TRUE.                                                              LINE 160
      IF(JSKIP.EQ.0) SYMB=.FALSE.                                              LINE 170
      IL=-2                                                                    LINE 180
      IF(JSKIP.LT.0) IL=-1                                                     LINE 190
      CALL PLOT((X(NA)-XMIN)/DX,(Y(NA)-YMIN)/DY,3)                             LINE 200
      DO 50 I=1,N                                                              LINE 210
      IF(SYMB.AND.MOD(NA-1,JMOD).EQ.0) GO TO 30                                LINE 220
      CALL PLOT((X(NA)-XMIN)/DX,(Y(NA)-YMIN)/DY,2)                             LINE 230
      GO TO 50                                                                 LINE 240
   30 CALL SYMBOL((X(NA)-XMIN)/DX,(Y(NA)-YMIN)/DY,HGT,L,0,IL)                  LINE 250
   50 NA=NA+ISKIP                                                              LINE 260
      RETURN                                                                   LINE 270
      END                                                                      LINE 280
```

SUBROUTINE PLTDAT

Description: Subroutine PLTDAT is used to identify plots. It is machine specific for the date and time software. The subroutine may be altered to reflect the form of plot identification desired (or a null subroutine may be used).

Subroutine listing:

```
      SUBROUTINE PLTDAT(X,Y)                                            PLTD  0
C     PLOTS DATE AND TIME FOR PLOT IDENTIFICATION                       PLTD  10
C     MACHINE SPECIFIC FOR DATE AND TIME SOFTWARE                       PLTD  20
      CALL SYMBOL(X,Y,.1,DATE(JULIAN),0.,10)                            PLTD  30
      CALL SYMBOL(999.,Y,.1,TIME(SECOND),0.,10)                         PLTD  40
      RETURN                                                            PLTD  50
      END                                                               PLTD  60
```

FUNCTION TIME

Description: FUNCTION TIME is a dummy function to substitute for the TIME function available on CDC 6000/7000 systems. If using such a system, FUNCTION TIME may be removed; for other systems it may be rewritten to properly access the system time, or it may be retained. It is called only from subroutine PLTDAT.

Function listing:

```
      FUNCTION TIME(ARG)                                                ME(A  0
C     DUMMY SUBROUTINE IF TIME NOT AVAILABLE                            ME(A  10
      DATA BLNK/1H /                                                    ME(A  20
      ARG=BLNK                                                          ME(A  30
      TIME=BLNK                                                         ME(A  40
      RETURN                                                            ME(A  50
      END                                                               ME(A  60
```

FUNCTION DATE

Description: FUNCTION DATE is a dummy function to substitute for the DATE function available on CDC 6000/7000 systems. As with FUNCTION TIME, FUNCTION DATE should be removed if using such a system and should be either rewritten or retained when used on other systems. It is called from subroutine PLTDAT and EDIT.

Function listing:

```
      FUNCTION DATE(ARG)                                                TE(A  0
C     DUMMY SUBROUTINE IF DATE NOT AVAILABLE                            TE(A  10
      DATA BLNK/1H /                                                    TE(A  20
      ARG=BLNK                                                          TE(A  30
      DATE=BLNK                                                         TE(A  40
      RETURN                                                            TE(A  50
      END                                                               TE(A  60
```

ASSEMBLER LANGUAGE SUBROUTINES

Since the program spends a large part of its time in matrix multiplication, the execution time may be reduced considerably by writing the two small matrix multiplication subroutines AMULT and SUMULT in efficient assembler language code. In the following listings these two subroutines are written in COMPASS for use on CDC systems. These particular subroutines should be usable on any 6000 or 7000 series CDC system with the FORTRAN EXTENDED compiler. (The RUN compiler has different subroutine linkage conventions.) The use of these subroutines in place of the FORTRAN routines will speed up the program by 20 percent to 25 percent. If extensive use on other systems is anticipated, it may be advisable to make assembler versions for them. Some FORTRAN optimizers may be efficient enough to negate the gain realized; the 20 percent to 25 percent improvement mentioned, however, is referenced to the highest level of optimization available with a CDC FORTRAN 4.0 compiler.

ASSEMBLY SUBROUTINE AMULT

Subroutine listing:

```
                                        IDENT  AMULT
                                        ENTRY  AMULT
                                        USE    CODE
                                        USE    /ALLDIM/
0                       1       MAX     BSS    1
1                       1       MAM     BSS    1
                                        USE CODE
0  0400400000 +                 AMULT   EQ     AMULT+4000008
1  5140000000 C                         SA4    MAX
              5150000001 C              SA5    MAM
2  63240                                SB2    X4              MAX
        63350                           SB3    X5              MAM
              6110777776                SB1    -1              -1
3  55211                                SA2    A1-B1           B
        55321                           SA3    A2-B1           C
              66421                     SB4    B2+B1           MAX-1
                    66631               SB6    B3+B1           MAM-1
4  53414                                SA4    X1+B4           A(MAX,1)
        53526                           SA5    X2+B6           B(MAM,1)
              10644                     BX6    X4
                    53634               SA6    X3+B4           =C(MAX,1)
5  26444                                UX4    B4,X4
        22444                           LX4    B4,X4
              63541                     SB5    X4+B1           II-1
                    26565               UX5    B6,X5
6  22565                                LX5    B6,X5
        63750                           SB7    X5              JJ
              54553                     SA5    A5+B3           B(MAM,2)
                    10655               BX6    X5
7  54662                                SA6    A6+B2           =C(MAX,2)
        26045                           UX0    B4,X5
              22040                     LX0    B4,X0           KK
                    66410               SB4    B1              I=0 FOR FIRST K
10 67441                        LOOPIK  SB4    B4-B1           I=I+1
        73714                           SX7    X1+B4
              66600                     SB6    B0              J=0
                    76600               SX6    B0
11 53470                        LOOPJ   SA4    X7              A
        53526                           SA5    X2+B6           B
              40445                     FX4    X4*X5           A*B
                    30664               FX6    X6+X4           +C
12 73772                                SX7    X7+B2           STEP A
        67661                           SB6    B6-B1           J=J+1
              0767000011 +              LT     B6,B7,LOOPJ
13 24606                                NX6    X6
        53634                           SA6    X3+B4           =C
              0745000010 +              LT     B4,B5,LOOPIK
14 66410                                SB4    B1              I=0
        73001                           SX0    X0+B1           K=K+1
              73223                     SX2    X2+B3           STEP B
                    73332               SX3    X3+B2           STEP C
15 0310000010 +                         NZ     X0,LOOPIK
              0400000000 +              EQ     AMULT
16                                      END
```

ASSEMBLY SUBROUTINE SUMULT

Subroutine listing:

```
                                                    IDENT SUMULT
                                                    ENTRY SUMULT
                                                    USE   /ALLDIM/
 0                          2      MAX              BSS   2
                                                    USE   CODE
 0  0400400000 +                   SUMULT           EQ    SUMULT+4000008
 1  6110000001                                       SB1   1              1
              5150000000 C                          SA5   MAX
 2  63250                                            SB2   X5
       10711                                        BX7   X1             XJI
              5011000001                            SA1   A1+1           XJID1
 3  5021000001                                       SA2   A1+1           SUM
              5042000001                            SA4   A2+1
 4  53340                                            SA3   X4
       63430                                        SB4   X3             JKM
              54441                                 SA4   A4+B1
                    53440                           SA4   X4
 5  42445                                            IX4   X4*X5
       63740                                        SB7   X4             MZ*MAX
              66500                                 SB5   B0
                    63351                           SB3   X5+B1          MAX+1
 6  73075                           LOOPIK           SX0   X7+B5          LOC(XJI(K,1))
       53525                                        SA5   X2+B5          SUM(K,I)
             10655                                  BX6   X5
                   66600                            SB6   B0             J=0
 7  53406                           LOOPJ            SA4   X0+B6          XJI(K,J)
       53516                                        SA5   X1+B6          XJI(I,J)
             40545                                  FX5   X4*X5
                   30656                            FX6   X5+X6
10  66662                                            SB6   B6+B2          J=J+1
       24606                                        NX6   X6
              0767000007 +                          LT    B6,B7,LOOPJ
11  53625                                            SA6   X2+B5          =SUM(I,K)
       66551                                        SB5   B5+B1          K=K+1
              0754000006 +                          LT    B5,B4,LOOPIK
12  66500                                            SB5   B0             RESTART K
       67441                                        SB4   B4-B1          LOWER K LIMIT   I=I+1
             73111                                  SX1   X1+B1          STEP LOC(XJI(I,1))
                   73223                            SX2   X2+B3          STEP SUM LOC TO DIAGONAL
13  73771                                            SX7   X7+B1
       0704000006 +                                 LT    B0,B4,LOOPIK
14  0400000000 +                                     EQ    SUMULT
15                                                   END
```

SEGMENTATION

Although the MMLE program does not require OVERLAY or SEGMENTATION to fit on most large computers, it is usually desirable to segment the program to decrease the load on the system. The following SEGMENTATION directives are used on the CDC OPERATING SYSTEM SCOPE 3.4 to reduce the loaded program size from $74,000_8$ words to $52,000_8$ words (including all buffers and system routines for input/output). The cost in execution time is negligible. The structure illustrated by these directives may be used as a guide for implementing the MMLE program on other systems.

```
PLTTREE    TREE      LINES-(THPLOT,APRPLT)
DATTREE    TREE      MATLD-(EDIT,DATA)
DOTREE     TREE      AEAT-(AGIRL,OUTPUT)
           TREE      MMLE-(PLTTREE,ASPIT-(DATTREE,DOTREE))
LINES      GLOBAL    LINCOM
ASPIT      GLOBAL    TOGIRL,INFO,TODATA,ROUTH,DIMENS
           GLOBAL    ALLDIM,BUF,MATRIX,COM,TOPLOT,HEADNG
           END
```

APPENDIX B

SAMPLE CHECK CASE FOR MMLE PROGRAM

This appendix presents a sample check case for the MMLE program. This listing is intended to aid the user in checking out the MMLE program; therefore, many of the available options have not been used.

INPUT CARDS

```
AIRCRAFT A CHECK CASE
 $INPUT CARD=T,Q=520.,V=4665., $END
  0 0 0   0   0 0 5875
A          4              4
-0.038      0.111        -1.0          0.0169
-16.79      -0.241        0.4          0.0
1.55        -0.00284     -0.042        0.0
              1.
B          4              5
             .0148
 12.76       20.08
 .3577       -2.445

01         7
 2160.       6.5          4869.        135.        22.65      2.7          198.
ENDCASE
  0 0 0  25      .7200      2.0000       .2808      .7999      .0150    -5.0024     -.2106
    -.7300       .0800      0.0000      0.0000
  0 0 0  50      .6800      2.0000       .2808      .7999      .0150    -5.0023     -.3495
    -.8000       .0800      0.0000      0.0000
  0 0 0  75      .6500      1.9996       .2922      .9002      .0150    -4.7503     -.5503
    -.8700       .1400      0.0000      0.0000
  0 0 0 100      .6300      1.5015       .2808      .9002      .0150    -4.2519     -.8020
    -.9200       .2200      0.0000      0.0000
  0 0 0 125      .6200      1.5012       .2521      .9002      .0150    -3.7474    -1.0487
    -.9750       .3500      0.0000      0.0000
  0 0 0 150      .6200      1.5015       .2292      .9002      .0150    -2.4985    -1.3984
   -1.0200       .6000      0.0000      0.0000
  0 0 0 175      .6200      1.5012       .2005      .9999      .0200    -1.2491    -1.7017
   -1.0650       .8500      0.0000      0.0000
  0 0 0 200      .6300      1.7476       .1490      .9999      .0230     1.0026    -2.0513
   -1.1100      1.0800      0.0000      0.0000
  0 0 0 225      .6500      1.9999       .0515      .9999      .0250     3.5008    -2.4522
   -1.1300      1.3300      0.0000      0.0000
  0 0 0 250      .7000      2.2979       0.0000     .9999      .0300     5.7451    -2.8991
   -1.1650      1.5500      0.0000      0.0000
  0 0 0 275      .7200      2.4984      -.0516      .9999      .0350     8.2507    -3.3007
   -1.1900      1.7400      0.0000      0.0000
  0 0 0 300      .7500      3.0022      -.1490     1.3001      .0420    11.9998    -3.7018
   -1.2200      1.9400      0.0000      0.0000
  0 0 0 325      .7800      3.5011      -.2292     1.3998      .0440    15.0034    -4.1483
   -1.2450      2.1000      0.0000      0.0000
  0 0 0 350      .8000      3.7989      -.3495     1.5001      .0440    16.5007    -4.5322
   -1.2700      2.2400      0.0000      0.0000
  0 0 0 375      .8100      4.2001      -.4527     1.5001      .0470    18.5029    -4.8476
   -1.2900      2.3700      0.0000      0.0000
  0 0 0 400      .8300      4.498       -.5672     1.6004      .0540    19.4973    -5.0995
   -1.3100      2.4700      0.0000      0.0000
  0 0 0 425      .8800      5.5005      -.6989     1.7001      .0540    20.0497    -5.4205
   -1.3400      2.5400      0.0000      0.0000
  0 0 0 450      .9200      5.9966      -.8194     1.8004      .0540    20.0510    -5.3002
   -1.3550      2.6600      0.0000      0.0000
  0 0 0 475      .9700      6.4995      -.9514     2.0003      .0540    20.0494    -5.4492
   -1.3650      2.6400      0.0000      0.0000
  0 0 0 500     1.0000      7.5012     -1.0487     2.2003      .0520    20.0510    -5.3002
   -1.3700      2.6500      0.0000      0.0000
  0 0 0 525     1.0300      7.9976     -1.2494     2.3001      .0520    19.9953    -5.1512
   -1.3800      2.6550      0.0000      0.0000
  0 0 0 550     1.1200      8.4944     -1.3808     2.5001      .0490    18.9998    -5.0024
   -1.3850      2.6400      0.0000      0.0000
  0 0 0 575     1.1800      9.4035     -1.5014     2.7001      .0440    17.9991    -4.8 15
   -1.390       2.6000      0.0000      0.0000
```

0 0 0 600	1.2300	9.5014	-1.6500	3.0002	.0400	15.9962	-4.5497
-1.4300	2.5600	0.0000	0.0000				
0 0 0 625	1.2800	9.9975	-1.7820	3.5005	.0300	13.5010	-4.2514
-1.4050	2.4800	0.0000	0.0000				
0 0 0 650	1.3500	10.1966	-1.9193	3.7004	.0230	9.9978	-3.9022
-1.4100	2.3800	0.0000	0.0000				
0 0 0 675	1.4000	10.4995	-1.9999	4.0001	.0200	6.2493	-3.4495
-1.4150	2.2600	0.0000	0.0000				
0 0 0 700	1.4500	10.8021	-2.0971	4.3004	-.0010	1.5014	-3.0486
-1.4150	2.1200	0.0000	0.0000				
0 0 0 725	1.5500	11.2024	-2.1778	4.5003	-.0090	-3.0025	-2.5500
-1.4150	1.9700	0.0000	0.0000				
0 0 0 750	1.6300	11.6014	-2.2173	4.7003	-.0160	-7.9986	-2.0513
-1.4150	1.7600	0.0000	0.0000				
0 0 0 775	1.7200	10.8002	-2.2905	5.0006	-.0330	-13.4995	-1.5013
-1.4150	1.5500	0.0000	0.0000				
0 0 0 800	1.8000	10.4960	-2.3208	5.3002	-.0640	-18.5019	-.9510
-1.4200	1.3400	0.0000	0.0000				
0 0 0 825	1.8800	10.1975	-2.3720	5.5002	-.0710	-23.7513	-.4524
-1.4230	1.0800	0.0000	0.0000				
0 0 0 850	1.9800	9.4984	-2.3777	5.9019	-.0880	-29.0006	.0516
-1.4250	.8400	0.0000	0.0000				
0 0 0 875	2.0200	8.4954	-2.3893	5.9993	-.1140	-33.9995	.4011
-1.4250	.6400	0.0000	0.0000				
0 0 0 900	2.1000	7.4985	-2.3781	5.9993	-.1360	-38.4998	.7506
-1.4250	.4800	0.0000	0.0000				
0 0 0 925	2.2000	6.4958	-2.3662	6.1999	-.1460	-41.4997	1.0027
-1.4250	.4100	0.0000	0.0000				
0 0 0 950	2.2300	5.2931	-2.3610	6.3039	-.1620	-44.4998	1.2492
-1.4250	.3400	0.0000	0.0000				
0 0 0 975	2.3200	3.9993	-2.3492	6.4978	-.1770	-46.4981	1.3984
-1.4230	.3000	0.0000	0.0000				
0 0 1 0	2.3900	2.8022	-2.2975	6.4978	-.1890	-48.4971	1.4782
-1.4200	.2600	0.0000	0.0000				
0 0 1 25	2.4300	1.5013	-2.2520	6.4978	-.1890	-50.4962	1.5529
-1.4150	.2500	0.0000	0.0000				
0 0 1 50	2.5200	.2980	-2.2002	6.4978	-.1960	-51.5039	1.6273
-1.4100	.2400	0.0000	0.0000				
0 0 1 75	2.6000	-1.0029	-2.1490	6.3030	-.2100	-53.0045	1.7018
-1.4000	.2200	0.0000	0.0000				
0 0 1 100	2.6300	-2.4985	-2.0969	6.3039	-.2200	-53.7489	1.7476
-1.4000	.2000	0.0000	0.0000				
0 0 1 125	2.6800	-3.7014	-1.9998	6.3039	-.2250	-54.4991	1.7991
-1.3900	.1900	0.0000	0.0000				
0 0 1 150	2.7200	-5.0022	-1.9481	6.3030	-.2290	-55.0028	1.8509
-1.3650	.1800	0.0000	0.0000				
0 0 1 175	2.7800	-6.4956	-1.9123	6.3030	-.2390	-55.5006	1.9193
-1.3400	.1750	0.0000	0.0000				
0 0 1 200	2.8000	-7.6984	-1.8510	6.1999	-.2340	-55.7462	1.9884
-1.3000	.1700	0.0000	0.0000				
0 0 1 225	2.8300	-9.0271	-1.7818	5.9993	-.2440	-55.9981	2.0112
-1.2600	.1650	0.0000	0.0000				
0 0 1 250	2.8500	-10.3046	-1.7475	5.7988	-.2460	-55.9980	2.0283
-1.2150	.1600	0.0000	0.0000				
0 0 1 275	2.8900	-11.5018	-1.7020	5.7002	-.2460	-55.7456	2.0514
-1.1600	.1550	0.0000	0.0000				
0 0 1 300	2.9000	-13.0026	-1.5989	5.0006	-.2480	-55.4990	2.0683
-1.1100	.1500	0.0000	0.0000				
0 0 1 325	2.9500	-15.0014	-1.5528	4.5003	-.2480	-55.0003	2.0801
-1.0500	.1500	0.0000	0.0000				
0 0 1 350	2.9700	-16.3016	-1.5011	4.3004	-.2510	-54.4962	2.0684
-.9900	.1500	0.0000	0.0000				
0 0 1 375	2.9700	-17.6987	-1.3979	3.5005	-.2480	-53.9977	2.0625
-.9250	.1500	0.0000	0.0000				
0 0 1 400	2.9700	-19.1998	-1.3692	3.0002	-.2480	-53.7457	2.0512
-.8650	.1500	0.0000	0.0000				

0 0 1 425	2.9500	-20.7978	-1.3006	2.5000	-.2480	-53.0010	2.0284
-.7900	.1500	0.0000	0.0000				
0 0 1 450	2.9300	-21.9947	-1.2322	2.0003	-.2480	-52.4968	2.0224
-.7300	.1500	0.0000	0.0000				
0 0 1 475	2.9200	-23.5012	-1.1805	1.5001	-.2460	-51.5000	1.9997
-.6700	.1500	0.0000	0.0000				
0 0 1 500	2.9000	-24.8013	-1.1173	.7999	-.2440	-49.9992	1.9481
-.6000	.1500	0.0000	0.0000				
0 0 1 525	2.8500	-26.0041	-1.0483	.3000	-.2440	-48.4983	1.9023
-.5400	.1500	0.0000	0.0000				
0 0 1 550	2.8000	-27.4991	-1.0025	-.5001	-.2390	-46.0004	1.8509
-.4700	.1500	0.0000	0.0000				
0 0 1 575	2.7500	-28.9999	-.9509	-.9999	-.2340	-42.4995	1.8220
-.4200	.1500	0.0000	0.0000				
0 0 1 600	2.6300	-30.5014	-.9681	-2.0003	-.2220	-39.5030	1.7819
-.3600	.1500	0.0000	0.0000				
0 0 1 625	2.5700	-31.8009	-.8482	-2.8003	-.2200	-36.2491	1.7518
-.3200	.1550	0.0000	0.0000				
0 0 1 650	2.4800	-33.1986	-.8023	-3.8001	-.2100	-33.4993	1.6271
-.2750	.1600	0.0000	0.0000				
0 0 1 675	2.3900	-34.4992	-.7792	-4.5003	-.2010	-30.0042	1.5817
-.2400	.1600	0.0000	0.0000				
0 0 1 700	2.3000	-35.4954	-.6992	-5.3002	-.1980	-26.9969	1.4785
-.2200	.1600	0.0000	0.0000				
0 0 1 725	2.2200	-36.4976	-.6478	-5.9993	-.1910	-24.0007	1.4098
-.2000	.1600	0.0000	0.0000				
0 0 1 750	2.1000	-37.7981	-.6190	-7.0021	-.1810	-21.9961	1.3178
-.1800	.1650	0.0000	0.0000				
0 0 1 775	2.0200	-39.0011	-.5786	-7.9991	-.1690	-19.0004	1.2319
-.1600	.1700	0.0000	0.0000				
0 0 1 800	1.9300	-39.5050	-.5503	-9.0018	-.1570	-16.5025	1.1519
-.1400	.1700	0.0000	0.0000				
0 0 1 825	1.8200	-40.5015	-.4984	-9.9988	-.1480	-13.7530	1.0485
-.1300	.1700	0.0000	0.0000				
0 0 1 850	1.7300	-40.9997	-.4587	-11.0016	-.1360	-11.2499	.9513
-.1200	.1700	0.0000	0.0000				
0 0 1 875	1.6300	-41.5036	-.4298	-11.9986	-.1290	-8.7473	.8311
-.1000	.1700	0.0000	0.0000				
0 0 1 900	1.5300	-41.5035	-.4183	-12.8008	-.1240	-6.2495	.7505
-.0900	.1700	0.0000	0.0000				
0 0 1 925	1.4300	-41.5032	-.3897	-13.4999	-.1050	-3.5012	.6477
-.0800	.1700	0.0000	0.0000				
0 0 1 950	1.3000	-40.9988	-.3784	-14.5026	-.0880	-1.2494	.5214
-.0600	.1700	0.0000	0.0000				
0 0 1 975	1.2200	-40.7982	-.3726	-15.8033	-.0810	1.0025	.4311
-.0500	.1700	0.0000	0.0000				
0 0 2 0	1.1200	-40.5001	-.3494	-16.5024	-.0640	3.5008	.2807
-.0400	.1700	0.0000	0.0000				
0 0 2 25	1.0100	-40.2993	-.3378	-17.2014	-.0520	6.0015	.1776
-.0250	.1700	0.0000	0.0000				
0 0 2 50	.9200	-39.8005	-.3208	-18.0036	-.0380	8.0001	.0516
-.0100	.1700	0.0000	0.0000				
0 0 2 75	.8000	-39.5023	-.2982	-19.0007	-.0300	9.9986	-.0802
0.0000	.1700	0.0000	0.0000				
0 0 2 100	.7000	-38.7975	-.2977	-20.0034	-.0180	11.9973	-.2177
.0100	.1700	0.0000	0.0000				
0 0 2 125	.6000	-38.0012	-.3211	-20.7999	-.0040	14.2533	-.3496
.0200	.1700	0.0000	0.0000				
0 0 2 150	.4700	-36.9985	-.3324	-21.4989	.0060	16.2523	-.4298
.0350	.1700	0.0000	0.0000				
0 0 2 175	.4600	-36.0016	-.3494	-22.5017	.0180	18.5027	-.5787
.0450	.1700	0.0000	0.0000				
0 0 2 200	.3000	-35.1997	-.3780	-23.5044	.0400	20.5013	-.6991
.0500	.1700	0.0000	0.0000				
0 0 2 225	.2200	-34.0022	-.3897	-24.0030	.0440	22.5001	-.8.79
.0600	.1700	0.0000	0.0000				

0 0 2 250	.1300	-33.3033	-.4127	-25.0000	.0520	24.4984	-.9509
.0650	.1700	0.0000	0.0000				
0 0 2 275	.0300	-32.0027	-.4300	-26.0027	.0640	25.5002	-1.0487
.0700	.1700	0.0000	0.0000				
0 0 2 300	-.0300	-30.7996	-.4526	-26.7992	.0730	27.4987	-1.1902
.0750	.1700	0.0000	0.0000				
0 0 2 325	-.1200	-29.4993	-.4986	-27.5040	.0860	28.9987	-1.3005
.0800	.1700	0.0000	0.0000				
0 0 2 350	-.1800	-28.5025	-.5213	-28.5013	.0980	30.9977	-1.3809
.0800	.1700	0.0000	0.0000				
0 0 2 375	-.2500	-27.5002	-.5670	-29.2001	.1070	31.9993	-1.5011
.0900	.1700	0.0000	0.0000				
0 0 2 400	-.3500	-26.3028	-.5901	-31.0023	.1190	33.4997	-1.5870
.0900	.1700	0.0000	0.0000				
0 0 2 425	-.4100	-25.5006	-.6016	-30.8045	.1290	34.2494	-1.6787
.0900	.1700	0.0000	0.0000				
0 0 2 450	-.4500	-23.9998	-.6474	-31.5035	.1360	35.4978	-1.7477
.0950	.1700	0.0000	0.0000				
0 0 2 475	-.5500	-22.9974	-.6817	-32.0021	.1410	36.2475	-1.8451
.1000	.1700	0.0000	0.0000				
0 0 2 500	-.5800	-21.6970	-.7219	-32.8042	.1480	36.9974	-1.9196
.1000	.1700	0.0000	0.0000				
0 0 2 525	-.6300	-20.5000	-.7794	-33.5033	.1560	37.4952	-1.9828
.1000	.1700	0.0000	0.0000				
0 0 2 550	-.6800	-19.6980	-.8023	-34.0018	.1600	38.4975	-2.0285
.1000	.1700	0.0000	0.0000				
0 0 2 575	-.7300	-18.4951	-.8479	-34.5003	.1650	38.7491	-2.0799
.0900	.1700	0.0000	0.0000				
0 0 2 600	-.7800	-17.3000	-.8994	-35.0045	.1700	39.0005	-2.1490
.0900	.1700	0.0000	0.0000				
0 0 2 625	-.8000	-15.9975	-.9227	-35.0045	.1700	39.5045	-2.1773
.0900	.1700	0.0000	0.0000				
0 0 2 650	-.8300	-14.5025	-.9800	-36.0016	.1730	40.0028	-2.2004
.0900	.1700	0.0000	0.0000				
0 0 2 675	-.8500	-13.5006	-1.0085	-36.3053	.1730	40.2489	-2.2289
.0901	.1700	0.0000	0.0000				
0 0 2 700	-.8800	-12.5033	-1.0484	-36.5001	.1820	40.5008	-2.2522
.0850	.1700	0.0000	0.0000				
0 0 2 725	-.9000	-11.5008	-1.0774	-36.7006	.1820	40.5009	-2.2407
.0800	.1700	0.0000	0.0000				
0 0 2 750	-.9000	-10.2008	-1.1515	-37.0043	.1850	40.7531	-2.2291
.0700	.1700	0.0000	0.0000				
0 0 2 775	-.9000	-9.5022	-1.1917	-37.2049	.1850	40.9995	-2.2289
.0600	.1700	0.0000	0.0000				
0 0 2 800	-.9000	-8.0011	-1.2203	-37.5028	.1850	40.9995	-2.2289
.0500	.1700	0.0000	0.0000				
0 0 2 825	-.9000	-6.9989	-1.2780	-37.6033	.1850	40.9997	-2.2174
.0350	.1700	0.0000	0.0000				
0 0 2 850	-.8800	-5.9966	-1.3180	-37.7034	.1850	40.9997	-2.2117
.0200	.1700	0.0000	0.0000				
0 0 2 875	-.850	-5.0024	-1.3898	-37.8038	.1850	40.9998	-2.2002
.0100	.1700	0.0000	0.0000				
0 0 2 900	-.8200	-3.5013	-1.3980	-37.8038	.1850	41.0001	-2.1773
-.0100	.1700	0.0000	0.0000				
0 0 2 925	-.8000	-2.8002	-1.4324	-37.9039	.1800	41.0004	-2.1486
-.0300	.1700	1.0000	0.0000				
0 0 2 950	-.7800	-1.5014	-1.4498	-37.9039	.1800	40.7543	-2.1202
-.0400	.1700	0.0000	0.0000				
0 0 2 975	-.7700	-.8024	-1.5185	-37.9039	.1800	40.4969	-2.0802
-.0500	.1700	0.0000	0.0000				
0 0 3 0	-.7500	.2981	-1.5527	-37.8038	.1750	39.9986	-2.0514
-.0600	.1700	0.0000	0.0000				
0 0 3 25	-.7000	1.0025	-1.5814	-37.8038	.1750	39.5005	-2.0111
-.0750	.1700	0.0000	0.0000				
0 0 3 50	-.6500	1.9997	-1.6213	-37.8038	.1730	38.4981	-1.9826
-.0750	.1700	0.0000	0.0000				

0 0 3 75	-.6300	3.0028	-1.6504	-37.7534	.1730	37.4959	-1.9198
-.0800	.1700	0.0000	0.0000				
0 0 3 100	-.6000	3.9996	-1.6788	-37.7034	.1700	36.4994	-1.8678
-.0850	.1700	0.0000	0.0000				
0 0 3 125	-.5500	4.6997	-1.7304	-37.5028	.1650	35.4972	-1.8107
-.0900	.1700	0.0000	0.0000				
0 0 3 150	-.5000	5.5007	-1.7476	-37.3023	.1600	33.4977	-1.7704
-.0950	.1700	0.0000	0.0000				
0 0 3 175	-.4300	6.2958	-1.7819	-37.0043	.1600	32.0029	-1.7217
-.1000	.1700	0.0000	0.0000				
0 0 3 200	-.3800	7.0004	-1.7992	-36.8038	.1480	30.5024	-1.6214
-.1000	.1700	0.0000	0.0000				
0 0 3 225	-.3300	7.7967	-1.8220	-36.7006	.1460	28.7493	-1.5815
-.1050	.1700	0.0000	0.0000				
0 0 3 250	-.2500	8.4955	-1.8507	-36.5031	.1410	27.0024	-1.5315
-.1100	.1700	0.0000	0.0000				
0 0 3 275	-.2000	9.2974	-1.8793	-36.3053	.1390	25.4962	-1.4212
-.1100	.1700	0.0000	0.0000				
0 0 3 300	-.1200	9.9963	-1.9023	-36.2021	.1270	23.9956	-1.3524
-.1150	.1700	0.0000	0.0000				
0 0 3 325	-.0500	10.5962	-1.9312	-36.0016	.1220	22.5008	-1.2780
-.1200	.1700	0.0000	0.0000				
0 0 3 350	0.0000	10.8037	-1.9482	-35.8010	.1170	21.0003	-1.1977
-.1230	.1700	0.0000	0.0000				
0 0 3 375	.0300	11.5026	-1.9711	-35.6005	.1100	19.4998	-1.1175
-.1240	.1700	0.0000	0.0000				
0 0 3 400	.1000	12.0011	-1.9770	-35.3025	.0980	17.9995	-1.0201
-.1250	.1700	0.0000	0.0000				
0 0 3 425	.1800	12.3047	-1.9826	-35.0045	.0950	16.2525	-.9509
-.1230	.1700	0.0000	0.0000				
0 0 3 450	.2500	12.7000	-1.9885	-34.5003	.0830	14.4999	-.8709
-.1200	.1700	0.0000	0.0000				
0 0 3 475	.3200	13.0038	-1.9883	-34.2998	.0760	13.4980	-.7794
-.1150	.1700	0.0000	0.0000				
0 0 3 500	.4000	13.5023	-1.9828	-34.0018	.0710	11.9977	-.6819
-.1100	.1700	0.0000	0.0000				
0 0 3 525	.4300	14.0009	-1.9824	-33.9044	.0640	10.4972	-.6016
-.1000	.1700	0.0000	0.0000				
0 0 3 550	.5300	14.3047	-1.9713	-33.8013	.0570	9.0027	-.4985
-.0900	.1700	0.0000	0.0000				
0 0 3 575	.6000	14.4021	-1.9712	-33.5033	.0490	7.5024	-.4211
-.0800	.1700	0.0000	0.0000				
0 0 3 600	.6500	14.5053	-1.9712	-33.0048	.0420	6.2541	-.3209
-.0700	.1700	0.0000	0.0000				
0 0 3 625	.7500	14.6028	-1.9596	-32.5035	.0400	5.0021	-.2806
-.0600	.1700	0.0000	0.0000				
0 0 3 650	.8300	14.6028	-1.9596	-31.9046	.0280	3.2486	-.0974
-.0400	.1700	0.0000	0.0000				
0 0 3 675	.9000	14.6946	-1.9480	-31.8015	.0200	1.7477	0.0000
-.0250	.1700	0.0000	0.0000				
0 0 3 700	.9800	14.5973	-1.9423	-31.5035	.0030	.2521	.0802
-.0100	.1700	0.0000	0.0000				
0 0 3 725	1.0300	14.5975	-1.9194	-31.0050	-.0010	-1.2493	.1777
.0100	.1700	0.0000	0.0000				
0 0 3 750	1.1200	14.5003	-1.9023	-30.5038	-.0090	-3.0027	.2693
.0250	.1700	0.0000	0.0000				
0 0 3 775	1.2000	14.3975	-1.8680	-30.3002	-.0140	-3.9997	.3782
.0500	.1700	0.0000	0.0000				
0 0 3 800	1.2700	14.3002	-1.8624	-30.0023	-.0330	-5.5008	.4526
.0700	.1700	0.0000	0.0000				
0 0 3 825	1.3500	13.9966	-1.8505	-29.8017	-.0380	-6.9960	.6016
.0900	.1700	0.0000	0.0000				
0 0 3 850	1.4000	13.6987	-1.8393	-29.5038	-.0470	-7.9980	.6821
.1200	.1700	0.0000	0.0000				
0 0 3 875	1.4700	13.5957	-1.8279	-29.3032	-.0570	-9.2459	.8020
.1400	.1700	0.0000	0.0000				

0 0 3 900	1.5200	13.4983	-1.8223	-28.9995	-.0620	-11.0041	.8994
.1700	.1700	0.0000	0.0000				
0 0 3 925	1.6000	13.0000	-1.7992	-28.5610	-.0710	-11.9997	1.0315
.2100	.1700	0.0000	0.0000				
0 0 3 950	1.6300	12.5017	-1.7818	-28.0025	-.0810	-13.4998	1.1518
.2300	.1700	0.0000	0.0000				
0 0 3 975	1.7000	12.3015	-1.7476	-27.8019	-.0860	-14.5015	1.2491
.2700	.1700	0.0000	0.0000				
0 0 4 0	1.7800	11.9983	-1.7019	-27.7645	-.1020	-15.9961	1.3522
.3000	.1700	0.0000	0.0000				
0 0 4 25	1.8300	11.5003	-1.6502	-27.5041	-.1070	-17.0029	1.5368
.3300	.1700	0.0000	0.0000				
0 0 4 50	1.8800	11.0023	-1.5984	-26.9997	-.1140	-17.9992	1.5816
.3500	.1700	0.0000	0.0000				
0 0 4 75	1.9300	10.8021	-1.5699	-26.5012	-.1190	-19.4998	1.6504
.3800	.1700	0.0000	0.0000				
0 0 4 100	1.9800	10.4991	-1.5314	-26.3007	-.1340	-20.5013	1.7706
.4000	.1700	0.0000	0.0000				
0 0 4 125	2.0100	9.5019	-1.5168	-26.2033	-.1340	-21.4976	1.8506
.4250	.1700	0.0000	0.0000				
0 0 4 150	2.0200	9.0046	-1.3983	-26.0027	-.1360	-22.4989	1.9828
.4500	.1700	0.0000	0.0000				
0 0 4 175	2.0500	8.4951	-1.3521	-25.8996	-.1430	-23.0025	2.0512
.4720	.1700	0.0000	0.0000				
0 0 4 200	2.0800	7.9974	-1.2723	-25.5042	-.1550	-23.5000	2.1488
.4900	.1700	0.0000	0.0000				
0 0 4 225	2.1000	7.4996	-1.1977	-24.7994	-.1550	-23.6997	2.2288
.5100	.1700	0.0000	0.0000				
0 0 4 250	2.1500	6.8.1.	-1.1518	-24.5015	-.1570	-23.8023	2.2804
.5300	.1700	0.0000	0.0000				
0 0 4 275	2.1700	6.4982	-1.0718	-24.3009	-.1650	-23.8987	2.3777
.5500	.1700	0.0000	0.0000				
0 0 4 300	2.1800	6.0004	-1.0028	-24.2035	-.1720	-23.9956	2.4182
.5700	.1700	0.0000	0.0000				
0 0 4 325	2.2100	5.0024	-.9222	-24.0030	-.1740	-23.9951	2.4698
.5800	.1700	0.0000	0.0000				
0 0 4 350	2.2200	4.4978	-.8481	-23.8998	-.1740	-23.8971	2.5209
.6000	.1700	0.0000	0.0000				
0 0 4 375	2.2300	3.9994	-.7791	-23.8024	-.1790	-23.7487	2.5325
.6100	.1700	0.0000	0.0000				
0 0 4 400	2.2400	3.5012	-.6993	-23.8024	-.1810	-23.7018	2.5727
.6220	.1700	0.0000	0.0000				
0 0 4 425	2.2500	3.0026	-.6017	-23.8024	-.1840	-23.5014	2.5499
.6400	.1700	0.0000	0.0000				
0 0 4 450	2.2700	2.2977	-.5501	-23.8024	-.1810	-22.9975	2.5210
.6500	.1700	0.0000	0.0000				
0 0 4 475	2.2800	1.7991	-.4527	-23.8024	-.1810	-22.7456	2.4983
.6600	.1700	0.0000	0.0000				
0 0 4 500	2.2900	1.1976	-.3782	-23.8024	-.1860	-22.4996	2.4527
.6700	.1700	0.0000	0.0000				
0 0 4 525	2.3000	.8021	-.2808	-23.8024	-.1860	-21.9958	2.4180
.6750	.1700	0.0000	0.0000				
0 0 4 550	2.3200	.2981	-.2305	-24.0030	-.1860	-21.4977	2.3778
.6900	.1700	0.0000	0.0000				
0 0 4 575	2.3000	0.0000	-.1318	-24.0030	-.1840	-20.5009	2.3493
.7000	.1700	0.0000	0.0000				
0 0 4 600	2.3000	-.4985	-.0688	-24.1004	-.1840	-19.7509	2.2806
.7050	.1700	0.0000	0.0000				
0 0 4 625	2.3000	-1.0026	.0172	-24.2035	-.1810	-19.0011	2.2003
.7100	.1700	0.0000	0.0000				
0 0 4 650	2.3000	-1.5014	.0516	-24.3009	-.1810	-17.9991	2.1317
.7200	.1700	0.0000	0.0000				
0 0 4 675	2.2800	-1.9997	.1203	-24.5015	-.1790	-17.5014	2.0513
.7220	.1700	0.0000	0.0000				
0 0 4 700	2.2700	-2.4980	.2005	-24.7020	-.1720	-16.7514	1.9826
.7240	.1700	0.0000	0.0000				

0 0 4 725	2.2500	-3.0025	.2292	-24.7994	-.1720	-15.9959	1.9023
.7300	.1700	0.0000	0.0000				
0 0 4 750	2.2000	-3.5011	.2808	-25.0000	-.1670	-14.7478	1.7990
.7350	.1750	0.0000	0.0000				
0 0 4 775	2.1500	-3.7989	.3495	-25.3037	-.1670	-14.2501	1.7192
.7400	.1750	0.0000	0.0000				
0 0 4 800	2.1200	-4.2975	.4011	-25.5042	-.1570	-13.5005	1.6217
.7450	.1750	0.0000	0.0000				
0 0 4 825	2.0700	-4.4983	.4298	-25.6016	-.1550	-12.5040	1.5698
.7500	.1750	0.0000	0.0000				
0 0 4 850	2.0300	-5.0020	.4813	-25.6990	-.1480	-12.0010	1.4498
.7500	.1750	0.0000	0.0000				
0 0 4 875	1.9800	-5.5006	.5271	-25.8022	-.1410	-11.1995	1.3811
.7500	.1750	0.0000	0.0000				
0 0 4 900	1.9500	-5.7987	.5673	-25.8996	-.1380	-9.9970	1.3006
.7500	.1750	0.0000	0.0000				
0 0 4 925	1.9000	-5.9989	.6016	-26.0027	-.1340	-9.0004	1.2492
.7500	.1750	0.0000	0.0000				
0 0 4 950	1.8500	-6.2967	.6188	-26.2033	-.1290	-7.9982	1.1978
.7550	.1750	0.0000	0.0000				
0 0 4 975	1.8300	-6.3996	.6472	-26.3007	-.1240	-7.0013	1.1803
.7600	.1750	0.0000	0.0000				
0 0 5 0	1.7500	-6.4966	.6821	-26.5012	-.1140	-6.2509	1.1516
.7630	.1750	0.0000	0.0000				
0 0 5 25	1.7000	-6.5993	.7222	-26.7992	-.1120	-5.5008	1.1001
.7650	.1750	0.0000	0.0000				
0 0 5 50	1.6500	-6.6963	.7679	-26.9023	-.1070	-4.2518	1.0770
.7680	.1750	0.0000	0.0000				
0 0 5 75	1.6000	-6.7990	.8022	-27.2003	-.1000	-3.0024	1.0487
.7700	.1750	0.0000	0.0000				
0 0 5 100	1.5500	-6.7989	.8194	-27.5040	-.0950	-1.5012	1.0315
.7700	.1750	0.0000	0.0000				
0 0 5 125	1.5200	-6.6956	.8309	-27.7045	-.0860	-.4985	1.0030
.7720	.1750	0.0000	0.0000				
0 0 5 150	1.4300	-6.6033	.8826	-27.8019	-.0810	1.0030	.9800
.7720	.1750	0.0000	0.0000				
0 0 5 175	1.4000	-6.5000	.8999	-28.0025	-.0760	2.4986	.9228
.7730	.1750	0.0000	0.0000				
0 0 5 200	1.3500	-6.2989	.9510	-28.0025	-.0670	3.9995	.8827
.7750	.1800	0.0000	0.0000				
0 0 5 225	1.3000	-6.0006	.9799	-28.2030	-.0620	5.9971	.8309
.7750	.1800	0.0000	0.0000				
0 0 5 250	1.2300	-5.7998	1.0030	-28.3034	-.0570	7.4983	.8194
.7800	.1800	0.0000	0.0000				
0 0 5 275	1.2000	-5.4949	1.0083	-28.4036	-.0570	9.3204	.8022
.7850	.1800	0.0000	0.0000				
0 0 5 300	1.1500	-5.2031	1.0315	-28.5010	-.0450	10.4953	.7908
.7880	.1800	0.0000	0.0000				
0 0 5 325	1.0800	-4.8017	1.0485	-28.7016	-.0350	11.4976	.7795
.7900	.1800	0.0000	0.0000				
0 0 5 350	1.0300	-3.9993	1.1172	-28.8047	-.0350	12.9987	.7508
.7900	.1800	0.0000	0.0000				
0 0 5 375	1.0000	-3.7989	1.1518	-28.9021	-.0280	14.5052	.6992
.7950	.1800	0.0000	0.0000				
0 0 5 400	.9800	-3.3006	1.1802	-28.9021	-.0230	15.5023	.6988
.8000	.1800	0.0000	0.0000				
0 0 5 425	.9200	-2.8019	1.1978	-28.9995	-.0210	16.5049	.6818
.8000	.1800	0.0000	0.0000				
0 0 5 450	.8800	-2.0000	1.2208	-28.9995	-.0180	18.0003	.6589
.8050	.1800	0.0000	0.0000				
0 0 5 475	.8300	-1.7019	1.2319	-29.1027	-.0140	18.7506	.6303
.8100	.1800	0.0000	0.0000				
0 0 5 500	.8000	-1.0026	1.2491	-28.9995	-.0060	20.2518	.6188
.8150	.1800	0.0000	0.0000				
0 0 5 525	.7500	-.4985	1.2780	-28.9021	-.0010	21.0023	.6016
.8170	.1800	0.0000	0.0000				

```
 0  0  5 550      .7200      .2006     1.3809  -28.8047      .0010   21.9992      .5789
      .8200      .1800     0.0000     0.0000
 0  0  5 575      .6800      .8024     1.3524  -28.7016      .0080   23.0017      .5498
      .8200      .1800     0.0000     0.0000
 0  0  5 600      .6500     1.5011     1.3695  -28.6041      .0180   24.2503      .4986
      .8230      .1800     0.0000     0.0000
 0  0  5 625      .6000     1.9998     1.3808  -28.5010      .0230   24.5020      .4524
      .8250      .1800     0.0000     0.0000
 0  0  5 650      .5700     2.4985     1.3984  -28.4036      .0280   25.4985      .4110
      .8260      .1800     0.0000     0.0000
 0  0  5 675      .5500     3.5010     1.4209  -28.3004      .0280   26.0022      .3498
      .8200      .1800     0.0000     0.0000
 0  0  5 700      .5000     3.9996     1.4328  -28.3004      .0350   26.5002      .2981
      .8300      .1800     0.0000     0.0000
 0  0  5 725      .4500     4.4986     1.4727  -28.2030      .0370   27.4968      .2522
      .8350      .1800     0.0000     0.0000
 0  0  5 750      .4300     5.3003     1.4784  -28.0025      .0420   27.7478      .1490
      .8370      .1800     0.0000     0.0000
 0  0  5 775      .4000     5.9985     1.5013  -27.8994      .0440   28.0051      .0974
      .8400      .1800     0.0000     0.0000
 0  0  5 800      .3500     6.4976     1.5527  -27.8019      .0520   28.5028      .0172
      .8400      .1800     0.0000     0.0000
 0  0  5 825      .3300     7.3000     1.5699  -27.7045      .0520   28.7023     -.0802
      .8400      .1800     0.0000     0.0000
 0  0  5 850      .3100     7.9993     1.5814  -27.6014      .0520   28.7474     -.1490
      .8450      .1800     0.0000     0.0000
 0  0  5 875      .2700     8.4978     1.5818  -27.5040      .0590   28.9987     -.2292
      .8500      .1800     0.0000     0.0000
AIRCRAFT B CHECK CASE
 $INPUT GROSWT=  2470.  ,IX=   275.  ,IY=  1902.  ,IZ=  2228.  ,IXZ=   11.6,
 Q=  39.0  ,V= 415.2  ,PUNCH=T  ,TIMESC= .5  ,BOTH=T,
 XALF=  0.00  ,XAN=  -.01  ,ZAX=   .53,
 ZMAX(3)=1000.,
 CARD=T,
 WMAPR= .10E+01  ,ALPHA=  7.86  ,MACH= .429  ,CG= .260  ,PARAM=    5.0000,
 LONG=T,  S=  85.  ,SPAN= 16.05  ,CBAR=  5.98  ,SPS= 5[illegible],  $END
113638750 113645840
A         4         4
   -.42335   1.00000  -0.00000    .00221
  -3.79430   -.36321   0.00000   1.00000
  15.66803   0.00000  -0.00000 -32.15869
   0.00000    .99155   0.00000   0.00000
B         4         5
   -.06489  -0.00000  -0.00000  -0.00000    .07346
  -6.28073   0.00000   0.00000   0.00000    .18165
   8.35392  -0.00000  -0.00000  -0.00000  -2.39365
            1.
01        5
  100000.0   70000.0       -0.   400000.0   10000.0       -0.0       -0.0
APRA      4         4
  .500E+05-0.        -0.        -0.
  .300E+00  .100E+04-0.        -0.
 0.        -0.        -0.        -0.
 0.        -0.        -0.        -0.
APRB      4         8
  .100E+05-0.        -0.        -0.        -0.        -0.        -0.        -0.
  .100E+01-0.        -0.        -0.        -0.        -0.        -0.        -0.
 0.        -0.        -0.        -0.        -0.        -0.        -0.        -0.
 0.        -0.        -0.        -0.        -0.        -0.        -0.        -0.
END
113638 765    8.5753   -1.9595  413.5412    4.3927     .7507    0.0001     -.0006
   -2.9244     .0309    -.0501    4.3839    2.054045599.9089     .4274   38.4406
113638 785    8.5616   -2.0726  413.6227    4.3253     .7578    0.0000     -.0005
   -2.3376     .0318    -.0486    4.3850    2.112545599.9089     .4274   38.4406
113638 805    8.5408   -2.2248  413.6921    4.3085     .7605    0.0000     -.0015
   -1.7396     .0322    -.0468    4.3858    2.196145599.9089     .4274   38.4406
```

113638 825	8.5509	-2.3516	413.7093	4. 89	.7521	0.0000	.0001
-1.5414	.0331	-.0456	4.3869	2 63545599.9089	.4274	38.4406	
113638 845	8.5872	-2.5260	413.7088	4.3038	.7402	0.0000	.0027
-1.9137	.0319	-.0437	4.3847	2.306645599.9089	.4274	38.4406	
113638 865	8.6012	-2.5961	413.7083	4.2496	.7307	0.0000	.0053
-2.6130	.0314	-.0411	4.3825	2.380245599.9089	.4274	38.4406	
113638 885	8.5797	-2.6251	413.7084	4.1654	.7241	0.0000	.0044
-3.3808	.0302	-.0386	4.3781	2.464545599.9089	.4274	38.4406	
113638 905	8.5756	-2.5501	413.7084	4.0980	.7197	0.0000	.0016
-4.1495	.0281	-.0343	4.3781	2.504545599.9089	.4274	38.4406	
113638 925	8.5627	-2.4277	413.7084	4.0813	.7094	0.0000	.0030
-4.9065	.0262	-.0314	4.3793	2.534945599.9089	.4274	38.4406	
113638 945	8.4809	-2.1846	413.7084	4.0555	.6899	0.0000	.0072
-5.6091	.0253	-.0288	4.3831	2.607945599.9089	.4274	38.4406	
113638 965	8.4024	-1.9271	413.7084	3.9824	.6769	0.0000	.0163
-6.075.	.0241	-.0256	4.3869	2.692645599.9089	.4274	38.4406	
113638 985	8.3974	-1.6396	413.7084	3.8979	.6845	0.0000	.0004
-6.0882	.0226	-.0223	4.3906	2.732745599.9089	.4274	38.4406	
113639 5	8.3924	-1.3778	413.6970	3.8580	.6937	0.0000	-.0042
-5.6172	.0212	-.0194	4.3935	2.763045599.9089	.4274	38.4406	
113639 25	8.3589	-1.178.	413.5372	3.8540	.7016	0.0000	-.0028
-4.8719	.0215	-.0178	4.3968	2.836045599.6841	.4269	38.3552	
113639 45	8.3762	-1.0928	413.1609	3.8549	.7085	0.0000	-.0006
-4.0660	.0214	-.0154	4.4030	2.920845583.6748	.4262	38.2710	
113639 65	8.4073	-1.0388	412.6949	3.8551	.7216	0.0000	.0001
-3.2368	.0213	-.0131	4.4120	2.960945583.5624	.4261	38.2283	
113639 85	8.4205	-1.0744	412.3642	3.8550	.7233	0.0000	-.0011
-2.3875	.0198	-.0101	4.4225	2.964945583.4500	.4258	38.1855	
113639 105	8.4568	-1.2403	412.2143	3.8550	.7297	0.0000	-.0023
-1.5454	.0186	-.0075	4.4328	2.969645583.4500	.4258	38.1855	
113639 125	8.4908	-1.4808	412.1725	3.8550	.7473	0.0000	-.0057
-.8791	.0164	-.0051	4.4407	3.023945583.4500	.4258	38.1855	
113639 145	8.4271	-1.7332	412.1700	3.8289	.7544	0.0000	-.0060
-.6036	.0154	-.0036	4.4463	3.108545583.4500	.4258	38.1855	
113639 165	8.3779	-2.0156	412.1710	3.7553	.7504	0.0000	-.0069
-.7918	.0129	-.0024	4.4493	3.176245583.4500	.4258	38.1855	
113639 185	8.4097	-2.2631	412.1710	3.6708	.7553	0.0000	-.0064
-1.3128	.0115	-.0017	4.4527	3.193045583.4500	.4258	38.1855	
113639 205	8.4246	-2.438.	412.1710	3.6310	.7624	0.0000	.0098
-2.0049	.0094	-.0012	4.4544	3.192645583.4500	.4258	38.1855	
113639 225	8.3743	-2.5167	412.1710	3.6008	.7494	0.0000	.0089
-2.7575	.0082	-.0007	4.4577	3.192145583.4500	.4258	38.1855	
113639 245	8.3433	-2.5156	412.1710	3.5281	.7272	0.0000	.0067
-3.5236	.0058	-.0002	4.4622	3.192145583.4500	.4258	38.1855	
113639 265	8.3208	-2.4164	412.1710	3.4438	.7124	0.0000	.0062
-4.2852	.0027	.0000	4.4671	3.192145583.4500	.4258	38.1855	
113639 285	8.2572	-2.2232	412.1710	3.4039	.6955	0.0000	.0049
-5.0012	.0000	-.0000	4.4721	3.192145583.4500	.4258	38.1855	
113639 305	8.2156	-2.0162	412.1710	3.3999	.6821	0.0000	.0037
-5.4495	-.0033	-.0005	4.4786	3.192145583.4500	.4258	38.1855	
113639 325	8.2278	-1.8144	412.1443	3.4008	.6723	0.0000	.0070
-5.3935	-.0054	-.0012	4.4856	3.192145583.4500	.4258	38.1855	
113639 345	8.2101	-1.6285	412.0635	3.4010	.6730	0.0000	.0084
-4.8517	-.0081	-.0024	4.4942	3.192145583.3376	.4255	38.1428	
113639 365	8.1986	-1.4795	411.9219	3.3748	.6858	0.0000	.0045
-4.0525	-.0097	-.0039	4.5040	3.192145583.2252	.4253	38.1000	
113639 385	8.2485	-1.4631	411.7953	3.3012	.7017	0.0000	.0014
-3.1842	-.0117	-.0059	4.5147	3.192145583.2252	.4253	38.1000	
113639 405	8.2806	-1.5141	411.7223	3.2168	.7179	0.0000	.0027
-2.2934	-.0138	-.0083	4.5241	3.192145583.2252	.4253	38.1000	
113639 425	8.2579	-1.6447	411.7062	3.1770	.7294	0.0000	.0013
-1.4101	-.0156	-.0110	4.5338	3.192145583.2252	.4253	38.1000	
113639 445	8.2446	-1.9136	411.7067	3.1729	.7409	0.0000	-.0055
-.5447	-.0172	-.0139	4.5430	3.186545583.2252	.4253	38.1000	
113639 465	8.2341	-2.2763	411.7072	3.1739	.7466	0.0000	-.0120
.2533	-.0184	-.0167	4.5499	3.132145583.2252	.4253	38.1000	

113639 485	8.1615	-2.6704	411.7072	3.1479	.7476	0.0000	-.0132
.8164	-.0189	-.0183	4.5525	3.047545583.2252		.4253	38.1000
113639 505	8.0815	-3.0977	411.7071	3.0743	.7420	0.0000	-.0102
.9448	-.0206	-.0210	4.5539	2.979845583.2252		.4253	38.1000
113639 525	8.0343	-3.5350	411.7071	2.9899	.7405	0.0000	-.0152
.6279	-.0217	-.0227	4.5542	2.963045583.2252		.4253	38.1000
113639 545	7.9862	-3.9021	411.7035	2.9239	.7331	0.0000	-.0104
.0335	-.0237	-.0253	4.5553	2.937245583.2252		.4253	38.1000
113639 565	7.9403	-4.1584	411.6851	2.8463	.7157	0.0000	.0022
-.6994	-.0239	-.0259	4.5546	2.863745567.4531		.4251	38.1013
113639 585	7.926[illegible]	-4.3669	411.6648	2.7629	.6994	0.0000	.0027
-1.4893	-.0247	-.0274	4.5549	2.778845583.2252		.4253	38.1000
113639 605	7.9117	-4.5001	411.6633	2.6971	.6879	0.0000	.0005
-2.2923	-.0258	-.0285	4.5530	2.738845583.2252		.4253	38.1000
113639 625	7.8604	-4.5177	411.6611	2.6195	.6722	0.0000	-.0011
-3.0865	-.0277	-.0306	4.5519	2.734745583.2252		.4253	38.1000
113639 645	7.7962	-4.4364	411.6338	2.5305	.6545	0.0000	-.0010
-3.8947	-.0292	-.0321	4.5496	2.709445567.4531		.4251	38.1013
113639 665	7.7513	-4.2546	411.5866	2.4367	.6334	0.0000	.0011
-4.6789	-.0303	-.0349	4.5497	2.635645567.4531		.4251	38.1013
113639 685	7.6943	-3.9591	411.5608	2.3486	.6104	0.0000	.0014
-5.4106	-.0303	-.0371	4.5495	2.550745567.4531		.4251	38.1013
113639 705	7.6482	-3.6214	411.5579	2.2767	.5961	0.0000	.0004
-5.8926	-.0312	-.0408	4.5504	2.510645567.4531		.4251	38.1013
113639 725	7.6520	-3.2974	411.5585	2.2060	.5999	0.0000	-.0013
-5.8928	-.0310	-.0418	4.5509	2.503945567.4531		.4251	38.1013
113639 745	7.6546	-3.0145	411.5586	2.1224	.6074	0.0000	-.0014
-5.3849	-.0300	-.0437	4.5551	2.447545567.4531		.4251	38.1013
113639 765	7.5991	-2.7744	411.5586	2.0557	.6126	0.0000	-.0010
-4.5873	-.0286	-.0445	4.5583	2.363145567.4531		.4251	38.1013
113639 785	7.5399	-2.6415	411.5585	2.0389	.6193	0.0000	-.0004
-3.7049	-.0277	-.0466	4.5630	2.295445567.4531		.4251	38.1013
113639 805	7.5089	-2.5788	411.5585	2.0337	.6309	0.0000	-.0009
-2.8041	-.0270	-.0473	4.5636	2.252445567.4531		.4251	38.1013
113639 825	7.4980	-2.6031	411.5585	1.9802	.6419	0.0000	-.0027
-1.8930	-.0261	-.0497	4.5647	2.178945567.4531		.4251	38.1013
113639 845	7.5116	-2.7503	411.5585	1.8962	.6489	0.0000	-.0045
-.9899	-.0252	-.0511	4.5638	2.094545567.4531		.4251	38.1013
113639 865	7.5187	-3.0047	411.5585	1.8289	.6612	0.0000	-.0048
-.1498	-.0242	-.0522	4.5645	2.028245567.4531		.4251	38.1013
113639 885	7.4768	-3.3096	411.5585	1.8066	.6693	0.0000	-.0048
.4637	-.0229	-.0509	4.5624	1.950245567.4531		.4251	38.1013
113639 905	7.4125	-3.6380	411.5585	1.7529	.6732	0.0000	-.0064
.6720	-.0222	-.0519	4.5601	1.866445567.4531		.4251	38.1013
113639 925	7.3834	-3.9659	411.5585	1.6695	.6648	0.0000	-.0086
.4108	-.0223	-.0502	4.5561	1.820945567.4531		.4251	38.1013
113639 945	7.3499	-4.2379	411.5415	1.5761	.6581	0.0000	-.0075
-.1821	-.0223	-.0499	4.5530	1.762445567.4531		.4251	38.1013
113639 965	7.2837	-4.4543	411.4934	1.4859	.6421	0.0000	-.0052
-.9383	-.0215	-.0471	4.5482	1.678945551.6928		.4250	38.1025
113639 985	7.2205	-4.5647	411.4381	1.3964	.6286	0.0000	-.0052
-1.7436	-.0209	-.0451	4.5450	1.611445551.6928		.4250	38.1025
113640 5	7.1776	-4.6089	411.4121	1.3831	.6168	0.0000	-.0075
-2.5729	-.0208	-.0423	4.5402	1.589045551.6928		.4250	38.1025
113640 25	7.1304	-4.5240	411.4094	1.2151	.5982	0.0000	-.0077
-3.4037	-.0201	-.0400	4.5358	1.535245551.6928		.4250	38.1025
113640 45	7.0858	-4.3477	411.4101	1.1432	.5712	0.0000	-.0042
-4.2229	-.0198	-.0366	4.5306	1.451145551.6928		.4250	38.1025
113640 65	7.0637	-4.0891	411.4101	1.0726	.5524	0.0000	-.0006
-5.0267	-.0185	-.0345	4.5258	1.383445551.6928		.4250	38.1025
113640 85	7.0516	-3.7457	411.4101	.9895	.5461	0.0000	-.0004
-5.7853	-.0182	-.0303	4.5204	1.366645551.6928		.4250	38.1025
113640 105	7.0292	-3.3494	411.4101	.9167	.5448	0.0000	-.0012
-6.2541	-.0181	-.0288	4.5169	1.361445551.6928		.4250	38.1025
113640 125	7.0075	-2.9264	411.3834	.8459	.5425	0.0000	-.0026
-6.1758	-.0168	-.0255	4.5126	1.307545551.6928		.4250	38.1025

113640 145	6.9644	-2.5579	411.2815	.7624	.5468	0.0000	-.0038
-5.5759	-.0156	-.0246	4.5116	1.223145551.5805		.4247	38.0598
113640 165	6.9296	-2.2471	411.1199	.6957	.5622	0.0000	-.0062
-4.7276	-.0142	-.0222	4.5107	1.155545551.4682		.4245	38.0170
113640 185	6.9509	-2.0761	410.9929	.6790	.5797	0.0000	-.0058
-3.8206	-.0138	-.0218	4.5109	1.138745551.4682		.4245	38.0170
113640 205	7.0185	-1.9898	410.9480	.6793	.5917	0.0000	-.0056
-2.8999	-.0136	-.0200	4.5082	1.139145551.4682		.4245	38.0170
113640 225	7.0708	-1.9887	410.9448	.6798	.5991	0.0000	-.0082
-1.9766	-.0134	-.0188	4.5064	1.139645551.4682		.4245	38.0170
113640 245	7.0887	-2.0849	410.9459	.6537	.6085	0.0000	-.0110
-1.0609	-.0127	-.0161	4.5038	1.139545551.4682		.4245	38.0170
113640 265	7.0668	-2.3011	410.9784	.5803	.6219	0.0000	-.0117
-.1970	-.0118	-.0137	4.5019	1.139545551.4682		.4245	38.0170
113640 285	7.0271	-2.5919	411.1147	.4960	.6297	0.0000	-.0112
.5241	-.0108	-.0102	4.4976	1.139545551.6928		.4250	38.1425
113640 305	7.0045	-2.9428	411.3422	.4506	.6306	0.0000	-.0102
.9987	-.0103	-.0074	4.4915	1.139545551.8051		.4252	38.1453
113640 325	6.9934	-3.3233	411.5377	.3925	.6290	0.0000	-.0097
1.1520	-.0098	-.0040	4.4841	1.139545551.8051		.4252	38.1453
113640 345	6.9488	-3.7194	411.6544	.3096	.6269	0.0000	-.0100
.9748	-.0098	-.0004	4.4768	1.139545551.8051		.4252	38.1453
113640 365	6.8709	-4.0653	411.7452	.2425	.6239	0.0000	-.0125
.5325	-.0098	.0037	4.4696	1.139545551.9174		.4255	38.1880
113640 385	6.8117	-4.3533	411.8037	.1997	.6112	0.0000	-.0124
-.0810	-.0099	.0076	4.4641	1.139545551.9174		.4255	38.1880
113640 405	6.7841	-4.5533	411.7955	.1266	.5916	0.0000	-.0104
-.7934	-.0094	.0113	4.4593	1.139545551.8051		.4252	38.1453
113640 425	6.7540	-4.6728	411.7772	.0155	.5742	0.0000	-.0081
-1.5786	-.0088	.0139	4.4541	1.139545551.9174		.4255	38.1880
113640 445	6.7146	-4.7068	411.8288	-.1012	.5646	0.0000	-.0092
-2.3973	-.0083	.0160	4.4465	1.139545536.2811		.4255	38.2320
113640 465	6.6734	-4.6335	411.8957	-.1936	.5614	0.0000	-.0104
-3.2418	-.0080	.0179	4.4380	1.139545536.2811		.4255	38.2320
113640 485	6.6236	-4.4591	411.9435	-.2617	.5500	0.0000	-.0093
-4.0836	-.0079	.0201	4.4288	1.165845536.2811		.4255	38.2320
113640 505	6.5801	-4.1750	411.9628	-.3427	.5320	0.0000	-.0067
-4.9155	-.0083	.0221	4.4227	1.239645536.2811		.4255	38.2320
113640 525	6.5553	-3.7833	412.0180	-.4301	.5152	0.0000	-.0054
-5.7217	-.0086	.0235	4.4186	1.324445536.3933		.4258	38.2748
113640 545	6.5146	-3.2856	412.1033	-.4774	.5031	0.0000	-.0056
-6.5017	-.0088	.0248	4.4140	1.390745536.3933		.4258	38.2748
113640 565	6.4414	-2.7306	412.1719	-.5383	.4961	0.0000	-.0059
-7.1179	-.0091	.0256	4.4084	1.468645536.3933		.4258	38.2748
113640 585	6.3939	-2.1577	412.1889	-.6253	.4950	0.0000	-.0074
-7.3263	-.0093	.0259	4.4027	1.552545536.3933		.4258	38.2748
113640 605	6.3828	-1.6004	412.1885	-.6955	.5001	0.0000	-.0100
-7.0376	-.0093	.0251	4.3983	1.618645536.3933		.4258	38.2748
113640 625	6.4021	-1.0919	412.1880	-.7131	.5031	0.0000	-.0102
-6.4054	-.0089	.0236	4.3945	1.696645536.3933		.4258	38.2748
113640 645	6.4546	-.6705	412.1881	-.7127	.5092	0.0000	-.0079
-5.6369	-.0088	.0226	4.3945	1.806745536.3933		.4258	38.2748
113640 665	6.5216	-.3391	412.1881	-.7121	.5204	0.0000	-.0071
-4.8121	-.0081	.0192	4.3961	1.920545536.3933		.4258	38.2748
113640 685	6.5717	-.1023	412.1881	-.7122	.5336	0.0000	-.0092
-3.9468	-.0074	.0163	4.3967	2.035745536.3933		.4258	38.2748
113640 705	6.5891	.0179	412.1881	-.7122	.5467	0.0000	-.0118
-3.0331	-.0061	.0124	4.3928	2.148745536.3933		.4258	38.2748
113640 725	6.5806	.0233	412.1881	-.7122	.5626	0.0000	-.0130
-2.1098	-.0051	.0094	4.3865	2.263745536.3933		.4258	38.2748
113640 745	6.5726	-.0782	412.1881	-.7122	.5816	0.0000	-.0131
-1.2149	-.0035	.0056	4.3805	2.376745536.3933		.4258	38.2748
113640 765	6.6124	-.2655	412.1881	-.7122	.5984	0.0000	-.0128
-.3748	-.0022	.0018	4.3784	2.491845536.3933		.4258	38.2748
113640 785	6.6960	-.5360	412.1881	-.7122	.6084	0.0000	-.0125
.3738	-.0009	-.0014	4.3784	2.604845536.3933		.4258	38.2748

113640 805	6.7303	-.8618	412.1938	-.7122	.6146	0.0000	-.0113
.9201	-.0015	-.0003	4.3775	2.719945536	.3933	.4258	38.2748
113640 825	6.7194	-1.2500	412.2452	-.7122	.6208	0.0000	-.0115
1.1400	-.0018	.0002	4.3717	2.833045536	.5054	.4260	38.3175
113640 845	6.7228	-1.6553	412.2991	-.7395	.6255	0.0000	-.0129
.9901	-.0013	-.0012	4.3629	2.948145520	.7686	.4259	38.3188
113640 865	6.7623	-2.0258	412.3576	-.8165	.6212	0.0000	-.0123
.5463	-.0001	-.0040	4.3514	3.066845536	.5054	.4263	38.3175
113640 885	6.7905	-2.3265	412.3894	-.9048	.6029	0.0000	-.0095
-.0906	-.0001	-.0045	4.3397	3.210145520	.8807	.4261	38.3615
113640 905	6.7796	-2.5559	412.4171	-.9464	.5861	0.0000	-.0089
-.8228	-.0007	-.0036	4.3312	3.339445536	.5054	.4260	38.3175
113640 925	6.7385	-2.6859	412.3830	-.9565	.5791	0.0000	-.0110
-1.5956	-.0021	-.0009	4.3293	3.465545520	.7686	.4259	38.3188
113640 945	6.6842	-2.6935	412.3332	-1.0122	.5739	0.0000	-.0119
-2.3868	-.0046	.0039	4.3282	3.592345520	.7686	.4259	38.3188
113640 965	6.6612	-2.6015	412.2853	-1.0999	.5627	0.0000	-.0101
-3.2144	-.0064	.0071	4.3226	3.719145520	.7686	.4259	38.3188
113640 985	6.6636	-2.4222	412.2704	-1.1704	.5470	0.0000	-.0065
-4.0647	-.0066	.0070	4.3139	3.894945520	.7686	.4259	38.3188
113641 5	6.6765	-2.1588	412.2703	-1.1879	.5308	0.0000	-.0030
-4.9040	-.0056	.0047	4.3071	4.045545520	.7686	.4259	38.3188
113641 25	6.6976	-1.8048	412.2708	-1.1875	.5210	0.0000	-.0020
-5.7156	-.0062	.0054	4.3022	4.196645520	.7686	.4259	38.3188
113641 45	6.7189	-1.3521	412.2708	-1.1870	.5178	0.0000	-.0046
-6.5041	-.0077	.0076	4.2992	4.352045520	.7686	.4259	38.3188
113641 65	6.7092	-.8386	412.2707	-1.2085	.5177	0.0000	-.0066
-7.1130	-.0097	.0106	4.2989	4.502645520	.7686	.4259	38.3188
113641 85	6.7159	-.2981	412.2708	-1.2288	.5183	0.0000	-.0063
-7.3151	-.0084	.0079	4.2979	4.653745520	.7686	.4259	38.3188
113641 105	6.7416	.2968	412.2708	-1.2292	.5234	0.0000	-.0064
-7.0355	-.0080	.0069	4.2964	4.809245520	.7686	.4259	38.3188
113641 125	6.7871	.6758	412.2708	-1.2005	.5335	0.0000	-.0070
-6.4287	-.0086	.0074	4.2949	4.959945520	.7686	.4259	38.3188
113641 145	6.8215	1.0615	412.2708	-1.1872	.5420	0.0000	-.0060
-5.6734	-.0107	.0098	4.2963	5.111245520	.7686	.4259	38.3188
113641 165	6.8603	1.3800	412.2671	-1.1866	.5560	0.0000	-.0058
-4.8624	-.0104	.0089	4.2992	5.266845520	.7686	.4259	38.3188
113641 185	6.9233	1.5838	412.2317	-1.1870	.5770	0.0000	-.0072
-4.0182	-.0093	.0069	4.3023	5.417545505	.0436	.4257	38.3200
113641 205	7.0201	1.6708	412.1768	-1.1870	.5961	0.0000	-.0063
-3.1255	-.0077	.0047	4.3042	5.563345505	.0436	.4257	38.3200
113641 225	7.1075	1.6673	412.1328	-1.1870	.6124	0.0000	-.0076
-2.2067	-.0084	.0053	4.3078	5.690745505	.0436	.4257	38.3200
113641 245	7.1524	1.5621	412.1484	-1.1870	.6310	0.0000	-.0060
-1.3078	-.0102	.0069	4.3133	5.810345505	.0436	.4257	38.3200
113641 265	7.1738	1.3415	412.2235	-1.1597	.6544	0.0000	-.0059
-.4664	-.0123	.0083	4.3191	5.953845505	.1556	.4260	38.3628
113641 285	7.1852	1.0207	412.3097	-1.0827	.6728	0.0000	-.0071
.2674	-.0129	.0084	4.3227	6.077945505	.1556	.4260	38.3628
113641 305	7.2175	.6295	412.3559	-.9944	.6821	0.0000	-.0090
.8024	-.0155	.0101	4.3242	6.176345505	.1556	.4260	38.3628
113641 325	7.2808	.2203	412.4151	-.9527	.6844	0.0000	-.0091
1.0262	-.0184	.0120	4.3243	6.266645505	.2676	.4262	38.4055
113641 345	7.3503	-.1925	412.4995	-.9485	.6818	0.0000	-.0079
.9075	-.0217	.0139	4.3235	6.377745505	.2676	.4262	38.4055
113641 365	7.3944	-.5932	412.5679	-.9554	.6746	0.0000	-.0068
.5577	-.0266	.0167	4.3227	6.491945505	.2676	.4262	38.4055
113641 385	7.4130	-.9764	412.6114	-1.0121	.6672	0.0000	-.0057
.1364	-.0323	.0199	4.3221	6.586945505	.2676	.4262	38.4055
113641 405	7.3926	-1.3062	412.6858	-1.1000	.6630	0.0000	-.0062
-.2622	-.0375	.0222	4.3229	6.681345505	.3796	.4264	38.4482
113641 425	7.3631	-1.5702	412.7719	-1.1704	.6595	0.0000	-.0082
-.6366	-.0413	.0233	4.3256	6.769945505	.3796	.4264	38.4482
113641 445	7.3630	-1.8011	412.8115	-1.1879	.6499	0.0000	-.0082
-1.0074	-.0459	.0245	4.3288	6.863545505	.3796	.4264	38.4482

113641 465	7.3764	-1.9779	412.8156	-1.1875	.6404	0.0000	-.0052
-1.4029	-.0514	.0261	4.3314	6.954745505.3796		.4264	38.4482
113641 485	7.3535	-2.0998	412.8147	-1.1870	.6335	0.0000	-.0028
-1.8586	-.0557	.0264	4.3350	7.039745505.3796		.4264	38.4482
113641 505	7.3049	-2.1635	412.8109	-1.2143	.6256	0.0000	-.0018
-2.3828	-.0602	.0271	4.3395	7.106945505.3796		.4264	38.4482
113641 525	7.2846	-2.1609	412.7890	-1.2913	.6174	0.0000	-.0009
-2.9436	-.0658	.0283	4.3437	7.184545489.6662		.4263	38.4495
113641 545	7.2856	-2.0969	412.7732	-1.3796	.6179	0.0000	-.0031
-3.5098	-.0731	.0296	4.3453	7.269245489.6662		.4263	38.4495
113641 565	7.2840	-1.9688	412.8041	-1.4213	.6196	0.0000	-.0047
-4.0621	-.0796	.0301	4.3469	7.314945489.7781		.4265	38.4922
113641 585	7.2826	-1.7841	412.8609	-1.4256	.6085	0.0000	-.0023
-4.5779	-.0852	.0300	4.3485	7.373845489.7781		.4265	38.4922
113641 605	7.2731	-1.5453	412.9141	-1.4246	.5961	0.0000	.0004
-5.0387	-.0915	.0295	4.3526	7.458045489.7781		.4265	38.4922
113641 625	7.2743	-1.2593	412.9644	-1.4303	.5931	0.0000	-.0014
-5.4322	-.0975	.0289	4.3583	7.526045489.7781		.4265	38.4922
113641 645	7.2945	-.9367	413.0386	-1.4869	.5899	0.0000	-.0033
-5.7521	-.1026	.0277	4.3648	7.543145489.8899		.4268	38.5350
113641 665	7.3238	-.5860	413.1019	-1.5749	.5906	0.0000	-.0024
-5.9762	-.1052	.0258	4.3695	7.548345489.8899		.4268	38.5350
113641 685	7.3381	-.2209	413.1844	-1.6453	.5898	0.0000	.0006
-6.0952	-.1091	.0235	4.3734	7.602645490.0018		.4270	38.5777
113641 705	7.3679	.1440	413.2731	-1.6628	.5929	0.0000	.0013
-6.1214	-.1138	.0210	4.3770	7.687845490.0018		.4270	38.5777
113641 725	7.4081	.4916	413.3408	-1.6566	.6024	0.0000	-.0011
-6.0699	-.1214	.0184	4.3821	7.756145490.0018		.4270	38.5777
113641 745	7.4376	.8247	413.3576	-1.5994	.6140	0.0000	-.0019
-5.9239	-.1275	.0156	4.3879	7.772945490.0018		.4270	38.5777
113641 765	7.4626	1.1440	413.3438	-1.5115	.6155	0.0000	.0009
-5.6691	-.1317	.0125	4.3941	7.772645490.0018		.4270	38.5777
113641 785	7.5164	1.4166	413.3135	-1.4411	.6206	0.0000	-.0018
-5.3267	-.1329	.0092	4.4000	7.772045490.0018		.4270	38.5777
113641 805	7.5690	1.6319	413.2652	-1.4236	.6372	0.0000	-.0007
-4.9299	-.1348	.0061	4.4059	7.772145474.3000		.4268	38.5790
113641 825	7.6108	1.8051	413.2279	-1.4241	.6554	0.0000	-.0021
-4.4981	-.1385	.0033	4.4109	7.772145474.3000		.4268	38.5790
113641 845	7.6597	1.9494	413.2102	-1.4245	.6673	0.0000	-.0015
-4.0433	-.1407	.0001	4.4161	7.772145474.3000		.4268	38.5790
113641 865	7.7488	2.0252	413.2079	-1.3971	.6761	0.0000	-.0016
-3.5742	-.1433	-.0028	4.4202	7.772145474.3000		.4268	38.5790
113641 885	7.8768	2.0335	413.2139	-1.3202	.6893	0.0000	-.0018
-3.0749	-.1471	-.0056	4.4239	7.772145474.3000		.4268	38.5790
113641 905	8.0000	1.9937	413.2688	-1.2319	.7006	0.0000	-.0012
-2.5602	-.1511	-.0079	4.4260	7.772145474.4118		.4271	38.6217
113641 925	8.0795	1.8834	413.3539	-1.1902	.7173	0.0000	-.0018
-2.0684	-.1544	-.0095	4.4265	7.772145474.4118		.4271	38.6217
113641 945	8.1186	1.6903	413.4277	-1.1859	.7306	0.0000	-.0024
-1.6253	-.1581	-.0103	4.4257	7.772145474.4118		.4271	38.6217
113641 965	8.1385	1.4291	413.4994	-1.1869	.7383	0.0000	-.0014
-1.2239	-.1619	-.0101	4.4246	7.772145474.5235		.4273	38.6644
113641 985	8.1499	1.1514	413.5841	-1.1870	.7452	0.0000	-.0029
-.8823	-.1642	-.0091	4.4223	7.772145474.5235		.4273	38.6644
113642 5	8.1768	.8657	413.6781	-1.1870	.7527	0.0000	-.0061
-.6458	-.1670	-.0075	4.4191	7.772145474.5235		.4273	38.6644
113642 25	8.2498	.5536	413.7695	-1.1870	.7614	0.0000	-.0075
-.5259	-.1727	-.0067	4.4169	7.772145474.6352		.4275	38.7071
113642 45	8.3543	.2099	413.8545	-1.1870	.7663	0.0000	-.0053
-.4995	-.1773	-.0066	4.4154	7.772145474.6352		.4275	38.7071
113642 65	8.4298	-.1315	413.8999	-1.1870	.7662	0.0000	-.0017
-.5424	-.1806	-.0056	4.4137	7.772145474.6352		.4275	38.7071
113642 85	8.4301	-.4597	413.9587	-1.1870	.7606	0.0000	.0021
-.6642	-.1833	-.0030	4.4104	7.772145474.7469		.4278	38.7498
113642 105	8.4094	-.7566	414.0427	-1.1870	.7552	0.0000	.0045
-.9032	-.1859	.0003	4.4065	7.772145474.7469		.4278	38.7498

113642 125	8.4327	-1.0202	414.1106	-1.1871	.7543	0.0000	.0162
-1.2675	-.1889	.0046	4.4021	7.772145474.7469		.4278	38.7498
113642 145	8.4780	-1.2382	414.1276	-1.1870	.7536	0.0000	.0171
-1.7173	-.1928	.0091	4.3981	7.777845474.7469		.4278	38.7498
113642 165	8.5037	-1.4136	414.1401	-1.1870	.7510	0.0000	.0162
-2.2184	-.1962	.0137	4.3938	7.811845474.7469		.4278	38.7498
113642 185	8.4932	-1.5163	414.1441	-1.1870	.7417	0.0000	.0021
-2.7542	-.1977	.0173	4.3996	7.877345474.8586		.4280	38.7925
113642 205	8.4739	-1.5403	414.2209	-1.2143	.7258	0.0000	-.0020
-3.3067	-.1994	.0208	4.3856	7.945245459.1683		.4279	38.7938
113642 225	8.4635	-1.4998	414.2239	-1.2913	.7102	0.0000	-.0026
-3.8601	-.2025	.0245	4.3822	7.990345459.1683		.4279	38.7938
113642 245	8.4669	-1.4259	414.2102	-1.3796	.7017	0.0000	.0005
-4.3958	-.2057	.0287	4.3792	8.002445459.1683		.4279	38.7938
113642 265	8.4733	-1.3015	414.2069	-1.4213	.7020	0.0000	.0030
-4.8840	-.2089	.0331	4.3767	8.002545459.1683		.4279	38.7938
113642 285	8.4709	-1.1238	414.2128	-1.4256	.7041	0.0000	.0046
-5.3120	-.2123	.0375	4.3743	8.002145459.1683		.4279	38.7938
113642 305	8.4743	-.8732	414.2675	-1.4245	.7003	0.0000	.0069
-5.6727	-.2136	.0416	4.3707	8.002145459.2799		.4281	38.8365
113642 325	8.4689	-.5903	414.3523	-1.4244	.7018	0.0000	.0078
-5.9532	-.2153	.0442	4.3671	8.028645459.2799		.4281	38.8365
113642 345	8.4717	-.2981	414.4202	-1.4245	.7082	0.0000	.0043
-6.1223	-.2195	.0466	4.3641	8.103245459.2799		.4281	38.8365
113642 365	8.5024	-.0091	414.4428	-1.4245	.7184	0.0000	-.0029
-6.1739	-.2237	.0494	4.3619	8.188945459.2799		.4281	38.8365
113642 385	8.5523	.2911	414.4969	-1.4245	.7200	0.0000	-.0058
-6.1194	-.2248	.0523	4.3595	8.229345459.3915		.4283	38.8792
113642 405	8.5822	.5724	414.6075	-1.4245	.7225	0.0000	-.0050
-5.9787	-.2230	.0534	4.3570	8.233445459.3915		.4283	38.8792
113642 425	8.5961	.8025	414.7459	-1.4245	.7167	0.0000	-.0004
-5.7561	-.2233	.0547	4.3546	8.232445459.5031		.4286	38.9219
113642 445	8.6294	1.0105	414.8160	-1.4245	.7149	0.0000	.0047
-5.4610	-.2247	.0562	4.3526	8.258845443.8243		.4284	38.9232
113642 465	8.6838	1.1922	414.8190	-1.4245	.7283	0.0000	.0065
-5.1137	-.2264	.0581	4.3511	8.333545459.5031		.4286	38.9219
113642 485	8.7506	1.3412	414.7854	-1.4245	.7503	0.0000	.0003
-4.7290	-.2272	.0599	4.3500	8.419245443.8243		.4284	38.9232
113642 505	8.8153	1.4179	414.7623	-1.4245	.7606	0.0000	-.0042
-4.2967	-.2261	.0632	4.3501	8.459645443.8243		.4284	38.9232
113642 525	8.8595	1.4284	414.7473	-1.3971	.7632	0.0000	-.0012
-3.8114	-.2244	.0607	4.3497	8.469445443.8243		.4284	38.9232
113642 545	8.8736	1.3996	414.7447	-1.3202	.7749	0.0000	.0014
-3.2801	-.2242	.0605	4.3489	8.523445443.8243		.4284	38.9232
113642 565	8.8811	1.3291	414.7450	-1.2319	.7905	0.0000	-.0020
-2.7438	-.2262	.0603	4.3480	8.608645443.8243		.4284	38.9232
113642 585	8.9093	1.1684	414.7507	-1.1902	.8002	0.0000	-.0052
-2.2467	-.2280	.0638	4.3468	8.677045443.8243		.4284	38.9232
113642 605	8.9563	.9046	414.8052	-1.1859	.8148	0.0000	-.0024
-1.7986	-.2285	.0639	4.3439	8.694045443.9358		.4287	38.9659
113642 625	9.0086	.5967	414.8954	-1.1869	.8382	0.0000	-.0017
-1.4014	-.2291	.0691	4.3390	8.720245443.9358		.4287	38.9659
113642 645	9.0580	.2816	415.0177	-1.1870	.8539	0.0000	-.0044
-1.0565	-.2297	.0731	4.3337	8.794445444.0473		.4289	39.0085
113642 665	9.0817	-.0565	415.1453	-1.1870	.8500	0.0000	-.0091
-.7822	-.2301	.0762	4.3297	8.880245444.0473		.4289	39.0085
113642 685	9.0864	-.4018	415.2866	-1.1870	.8453	0.0000	-.0131
-.5940	-.2293	.0791	4.3267	8.926445444.1587		.4291	39.0512
113642 705	9.0715	-.7777	415.3878	-1.1870	.8474	0.0000	-.0153
-.5046	-.2320	.0841	4.3231	8.985545444.1587		.4291	39.0512
113642 725	9.0385	-1.1860	415.4275	-1.2143	.8497	0.0000	-.0133
-.5125	-.2363	.0889	4.3193	9.070045444.1587		.4291	39.0512
113642 745	9.0044	-1.5860	415.4574	-1.2913	.8390	0.0000	-.0067
-.6304	-.2399	.0928	4.3160	9.144045444.1587		.4291	39.0512
113642 765	8.9872	-1.9502	415.5267	-1.3796	.8201	0.0000	.0010
-.8551	-.2421	.0966	4.3119	9.216145444.2702		.4294	39.0939

113642 785	8.9917	-2.2720	415.5815	-1.4213	.7939	0.0003	.0092
-1.1282	-.2443	.0998	4.3087	9.301245428.6029		.4292	39.0952
113642 805	8.9748	-2.5674	415.6207	-1.4256	.7839	0.0000	.0102
-1.3710	-.2447	.1029	4.3060	9.374945428.7142		.4294	39.1379
113642 825	8.9304	-2.8328	415.6650	-1.4519	.8027	0.0000	.0016
-1.5447	-.2451	.1061	4.3024	9.447145428.7142		.4294	39.1379
113642 845	8.8721	-3.0945	415.7207	-1.5287	.8219	0.0000	-.0048
-1.6749	-.2469	.1105	4.2966	9.532245428.7142		.4294	39.1379
113642 865	8.8340	-3.3047	415.7376	-1.6171	.8072	0.0000	-.0018
-1.7944	-.2487	.1137	4.2924	9.606045428.7142		.4294	39.1379
113642 885	8.7901	-3.4851	415.7638	-1.6646	.7868	0.0000	.0046
-1.9100	-.2507	.1174	4.2888	9.678245428.7142		.4294	39.1379
113642 905	8.7284	-3.6448	415.8371	-1.7255	.7867	0.0000	.0017
-2.0297	-.2538	.1212	4.2862	9.763445428.8256		.4297	39.1805
113642 925	8.6786	-3.7946	415.9217	-1.8125	.7915	0.0003	-.0060
-2.1759	-.2559	.1246	4.2840	9.858145428.8256		.4297	39.1805
113642 945	8.6489	-3.9164	415.9617	-1.9101	.7806	0.0000	-.0040
-2.3432	-.2573	.1262	4.2829	9.950245428.8256		.4297	39.1805
113642 965	8.6181	-4.0079	415.9658	-2.0047	.7609	0.0003	.0062
-2.5246	-.2575	.1273	4.2790	10.055745428.8256		.4297	39.1805
113642 985	8.5761	-4.0877	415.9910	-2.0926	.7456	0.0000	.0091
-2.7130	-.2594	.1299	4.2744	10.151645428.8256		.4297	39.1805
113643 6	8.5341	-4.1368	416.0647	-2.1611	.7368	0.0000	.0043
-2.9128	-.2586	.1329	4.2703	10.225545428.9369		.4299	39.2232
113643 25	8.4928	-4.1579	416.1322	-2.2424	.7215	0.0003	.0016
-3.1324	-.2619	.1373	4.2658	10.309745428.9369		.4299	39.2232
113643 45	8.4453	-4.1210	416.1295	-2.3356	.7055	0.0000	.0029
-3.3670	-.2648	.1415	4.2592	10.424445413.2812		.4298	39.2245
113643 65	8.3790	-4.0680	416.1323	-2.4339	.6983	0.0003	.0046
-3.5917	-.2689	.1457	4.2522	10.566745413.3924		.4300	39.2672
113643 86	8.2822	-3.9925	416.1894	-2.5261	.7053	0.0003	.0030
-3.7884	-.2691	.1474	4.2473	10.675045413.3924		.4300	39.2672
113643 105	8.1665	-3.8884	416.2540	-2.6015	.7111	0.0000	.0006
-3.9760	-.2695	.1484	4.2439	10.770145413.3924		.4300	39.2672
113643 126	8.0742	-3.7383	416.2715	-2.6755	.7077	0.0003	-.0011
-4.1668	-.2711	.1503	4.2414	10.885645413.3924		.4300	39.2672
113643 145	8.0266	-3.5806	416.2711	-2.7632	.6880	0.0000	-.0025
-4.3535	-.2764	.1541	4.2388	11.027945413.3924		.4303	39.2672
113643 166	7.9986	-3.4136	416.2706	-2.8390	.6713	0.0003	-.0030
-4.5233	-.2816	.1566	4.2354	11.157245413.3924		.4303	39.2672
113643 186	7.9579	-3.2287	416.2706	-2.9132	.6645	0.0000	-.0024
-4.6686	-.2848	.1584	4.2302	11.277945413.3924		.4302	39.2672
113643 206	7.9225	-3.0259	416.2968	-3.0009	.6657	0.0000	.0009
-4.7852	-.2857	.1586	4.2241	11.422645413.3924		.4300	39.2672
113643 226	7.8973	-2.8001	416.3704	-3.0709	.6566	0.0000	.0030
-4.8852	-.2850	.1562	4.2193	11.553545413.5036		.4302	39.3098
113643 246	7.8846	-2.5635	416.4811	-3.0884	.6507	0.0003	.0035
-4.9740	-.2856	.1539	4.2158	11.681245413.5036		.4302	39.3098
113643 266	7.8888	-2.3219	416.5774	-3.1154	.6526	0.0003	.0028
-5.0448	-.2878	.1524	4.2123	11.788745413.6148		.4305	39.3525
113643 286	7.8940	-2.0681	416.6176	-3.1919	.6581	0.0003	.0013
-5.0807	-.2889	.1492	4.2104	11.912145397.9706		.4303	39.3538
113643 306	7.8705	-1.8060	416.6272	-3.2803	.6616	0.0000	-.0009
-5.0882	-.2902	.1453	4.2104	12.041045397.9706		.4303	39.3538
113643 326	7.8488	-1.5385	416.6784	-3.3221	.6581	0.0000	-.0021
-5.0656	-.2911	.1399	4.2109	12.184545398.0817		.4305	39.3965
113643 346	7.8491	-1.2977	416.7592	-3.3264	.6498	0.0000	-.0008
-5.0227	-.2897	.1333	4.2115	12.313945398.0817		.4305	39.3965
113643 365	7.8505	-1.0682	416.7995	-3.3254	.6492	0.0003	.0017
-4.9533	-.2854	.1251	4.2133	12.429445398.0817		.4305	39.3965
113643 386	7.8355	-.8561	416.8037	-3.3252	.6630	0.0003	.0006
-4.8506	-.2832	.1176	4.2163	12.525145398.0817		.4305	39.3965
113643 406	7.8321	-.6442	416.8027	-3.3253	.6718	0.0000	-.0013
-4.7027	-.2821	.1102	4.2199	12.620245398.0817		.4305	39.3965
113643 426	7.8598	-.4432	416.8026	-3.3253	.6663	0.0001	-.0018
-4.5438	-.2833	.1033	4.2240	12.715945398.0817		.4305	39.3965

113643 446	7.914[illegible]	-.2701	416.7892	-3.3468	.6644	0.0000	-.0011
-4.3822	-.2851	.0967	4.2272	12.839345398.0817		.4305	39.3965
113643 466	7.9515	-.1424	416.7820	-3.3945	.6703	0.0000	-.0023
-4.2160	-.2843	.0896	4.2295	12.942345398.0817		.4305	39.3965
113643 486	7.9386	-.0362	416.8322	-3.4729	.6804	0.0000	-.0041
-4.0384	-.2819	.0827	4.2307	13.016645398.1928		.4308	39.4391
113643 506	7.9234	.0667	416.8986	-3.5316	.6909	0.0000	-.0048
-3.8601	-.2813	.0761	4.2329	13.088745382.5602		.4306	39.4404
113643 526	7.9502	.143[illegible]	416.9190	-3.5601	.6989	0.0000	-.0015
-3.6712	-.2799	.0695	4.2363	13.174445382.5602		.4306	39.4404
113643 546	7.9891	.1706	416.9123	-3.5638	.7040	0.0000	.0027
-3.4702	-.2773	.0631	4.2404	13.243045382.5602		.4306	39.4404
113643 566	8.0057	.1708	416.9461	-3.5632	.7125	0.0000	.0019
-3.2599	-.2768	.0582	4.2437	13.286945382.5602		.4306	39.4404
113643 586	8.0128	.1609	417.0141	-3.5631	.7237	0.0000	-.0023
-3.0412	-.2783	.0536	4.2460	13.362045382.6712		.4309	39.4831
113643 606	8.0121	.1356	417.0433	-3.5631	.7328	0.0000	-.0048
-2.8244	-.2801	.0488	4.2481	13.448145382.5602		.4306	39.4404
113643 626	8.0064	.0785	417.0599	-3.5631	.7392	0.0000	-.0030
-2.6166	-.2813	.0441	4.2515	13.489145382.6712		.4309	39.4831
113643 646	8.0214	-.0166	417.1065	-3.5631	.7403	0.0000	-.0017
-2.4270	-.2811	.0389	4.2562	13.493245382.6712		.4309	39.4831
113643 666	8.0563	-.1266	417.2024	-3.5631	.7385	0.0000	-.0005
-2.2538	-.2816	.0331	4.2609	13.519145382.7822		.4311	39.5257
113643 686	8.0812	-.2565	417.2909	-3.5631	.7386	0.0000	-.0014
-2.0998	-.2831	.0281	4.2636	13.594445382.7822		.4311	39.5257
113643 706	8.1004	-.4051	417.3302	-3.5631	.7412	0.0000	-.0040
-1.9635	-.2821	.0241	4.2650	13.681145382.7822		.4311	39.5257
113643 726	8.1229	-.5715	417.3341	-3.5631	.7412	0.0000	-.0055
-1.8459	-.2779	.0199	4.2676	13.722045382.7822		.4311	39.5257
113643 746	8.1375	-.7432	417.3160	-3.5631	.7361	0.0000	-.0028
-1.7546	-.2740	.0149	4.2724	13.726245382.7822		.4311	39.5257
113643 766	8.1174	-.9137	417.2676	-3.5631	.7380	0.0000	.0005
-1.6966	-.2715	.0098	4.2782	13.725245367.1611		.4309	39.5270
113643 786	8.0898	-1.0875	417.2382	-3.5631	.7477	0.0000	.0014
-1.6725	-.2721	.0055	4.2831	13.725145367.1611		.4309	39.5270
113643 806	8.0847	-1.2628	417.2853	-3.5631	.7547	0.0000	-.0005
-1.6816	-.2745	.0013	4.2868	13.725145367.2720		.4312	39.5697
113643 826	8.0918	-1.4487	417.3668	-3.5690	.7564	0.0000	-.0010
-1.7071	-.2755	-.0027	4.2898	13.725145367.2720		.4312	39.5697
113643 846	8.0822	-1.6231	417.4331	-3.6257	.7517	0.0000	-.0004
-1.7318	-.2727	-.0060	4.2925	13.725145367.2720		.4312	39.5697
113643 866	8.071[illegible]	-1.7938	417.5068	-3.7138	.7471	0.0000	.0006
-1.7689	-.2705	-.0093	4.2952	13.725145367.3829		.4314	39.6123
113643 886	8.0609	-1.9511	417.5938	-3.7843	.7425	0.0000	-.0003
-1.8557	-.2677	-.0126	4.2979	13.725145351.7733		.4313	39.6137
113643 906	8.0472	-2.0939	417.6641	-3.8019	.7361	0.0000	-.0009
-1.9897	-.2687	-.0157	4.3002	13.725145351.8841		.4315	39.6563
113643 926	8.0446	-2.2105	417.7081	-3.8014	.7205	0.0000	-.0014
-2.1465	-.2711	-.0184	4.3022	13.725145351.8841		.4315	39.6563
113643 946	8.0449	-2.3117	417.7178	-3.8283	.7151	0.0000	-.0027
-2.3157	-.2751	-.0220	4.3054	13.725145351.8841		.4315	39.6563
113643 966	8.0263	-2.3863	417.7192	-3.9054	.7156	0.0000	-.0034
-2.4960	-.2759	-.0256	4.3091	13.725145351.8841		.4315	39.6563
113643 986	8.0122	-2.4362	417.7720	-3.9940	.7201	0.0000	-.0028
-2.6817	-.2762	-.0294	4.3121	13.725145351.9949		.4317	39.6989
113644 6	7.9861	-2.4527	417.8557	-4.0357	.7169	0.0000	-.0032
-2.8846	-.2762	-.0330	4.3144	13.725145351.9949		.4317	39.6989
113644 26	7.9602	-2.4544	417.9226	-4.0459	.7135	0.0000	-.0050
-3.1223	-.2752	-.0365	4.3171	13.725145351.9949		.4317	39.6989
113644 46	7.9221	-2.4344	417.9652	-4.1016	.7054	0.0000	-.0035
-3.3754	-.2727	-.0397	4.3203	13.719445351.9949		.4317	39.6989
113644 66	7.8860	-2.3861	418.0379	-4.1896	.6981	0.0000	-.0005
-3.6219	-.2704	-.0424	4.3230	13.684945352.1057		.4320	39.7416
113644 86	7.8754	-2.3005	418.1212	-4.2603	.6923	0.0000	.0014
-3.8434	-.2690	-.0447	4.3260	13.618545352.1057		.4320	39.7416

113644 106	7.8763	-2.1989	418.1608	-4.2778	.6875	0.0000	.0017
-4.0354	-.2665	-.0460	4.3286	13.549845352.	.1057	.4320	39.7416
113644 126	7.8742	-2.0731	418.1908	-4.2833	.6859	0.0000	.0002
-4.2023	-.2631	-.0481	4.3313	13.504445352.	.1057	.4325	39.7416
113644 146	7.8627	-1.9258	418.2198	-4.3395	.6820	0.0000	-.0010
-4.3532	-.2634	-.0512	4.3339	13.491845352.	.2164	.4322	39.7842
113644 166	7.8642	-1.7417	418.1821	-4.4277	.6761	0.0000	-.0024
-4.5025	-.2622	-.0541	4.3365	13.491745336.	.5076	.4318	39.7429
113644 186	7.8728	-1.5301	418.1584	-4.4984	.6691	0.0000	-.0020
-4.6512	-.2594	-.0564	4.3401	13.465345336.	.5076	.4318	39.7429
113644 206	7.8726	-1.3112	418.1197	-4.5159	.6701	0.0000	-.0011
-4.7873	-.2572	-.0588	4.3441	13.389845336.	.6183	.4320	39.7855
113644 226	7.8555	-1.0926	418.1959	-4.5155	.6764	0.0000	-.0015
-4.8829	-.2542	-.0612	4.3484	13.303245336.	.6183	.4320	39.7855
113644 246	7.8344	-.8775	418.2424	-4.5150	.6764	0.0000	-.0028
-4.9483	-.2525	-.0628	4.3508	13.262445336.	.6183	.4320	39.7855
113644 266	7.8315	-.6505	418.2800	-4.5150	.6706	0.0000	-.0023
-4.9972	-.2503	-.0643	4.3524	13.231445336.	.7289	.4323	39.8281
113644 286	7.8584	-.4277	418.3431	-4.5150	.6656	0.0000	-.0012
-5.0425	-.2505	-.0660	4.3530	13.197145336.	.7289	.4323	39.8281
113644 306	7.8957	-.2008	418.4093	-4.5150	.6658	0.0000	-.0003
-5.0603	-.2487	-.0675	4.3538	13.079645336.	.7289	.4323	39.8281
113644 326	7.9108	.0068	418.4551	-4.5150	.6690	0.0000	-.0013
-5.0456	-.2425	-.0676	4.3553	13.029845336.	.7289	.4323	39.8281
113644 346	7.9142	.2055	418.4853	-4.5150	.6734	0.0000	-.0027
-5.0028	-.2339	-.0672	4.3583	12.998845321.	.2529	.4323	39.8721
113644 366	7.9283	.4036	418.5135	-4.5150	.6790	0.0000	-.0018
-4.9322	-.2287	-.0676	4.3612	12.924545321.	.2529	.4323	39.8721
113644 386	7.9631	.6053	418.5615	-4.5150	.6816	0.0000	.0029
-4.8309	-.2265	-.0694	4.3634	12.838245321.	.2529	.4323	39.8721
113644 406	8.0019	.7704	418.6399	-4.5150	.6859	0.0000	.0049
-4.7028	-.2241	-.0705	4.3637	12.791645321.	.3635	.4326	39.9147
113644 426	8.0375	.9089	418.7234	-4.5150	.7004	0.0000	.0018
-4.5480	-.2208	-.0711	4.3639	12.732145321.	.3635	.4326	39.9147
113644 446	8.0584	1.0327	418.7627	-4.5150	.7144	0.0000	-.0020
-4.3716	-.2189	-.0721	4.3650	12.647145321.	.3635	.4326	39.9147
113644 466	8.0834	1.1445	418.7722	-4.5150	.7195	0.0000	-.0014
-4.1841	-.2178	-.0731	4.3683	12.578445321.	.3635	.4326	39.9147
113644 486	8.1095	1.2291	418.8249	-4.5150	.7209	0.0000	.0011
-3.9991	-.2179	-.0741	4.3693	12.534545321.	.4741	.4328	39.9573
113644 506	8.1298	1.2779	418.9079	-4.5150	.7319	0.0000	.0019
-3.8101	-.2152	-.0746	4.3702	12.459745321.	.4741	.4328	39.9573
113644 526	8.1556	1.3015	418.9746	-4.5091	.7485	0.0000	.0003
-3.6144	-.2087	-.0731	4.3722	12.374045321.	.4741	.4328	39.9573
113644 546	8.2057	1.2936	418.9966	-4.4523	.7615	0.0000	-.0006
-3.4118	-.1996	-.0704	4.3742	12.306645321.	.4741	.4328	39.9573
113644 566	8.2889	1.2642	419.0189	-4.3642	.7677	0.0000	.0017
-3.2006	-.1902	-.0678	4.3757	12.227345321.	.5846	.4331	39.9999
113644 586	8.4081	1.1961	419.0145	-4.2936	.7634	0.0000	.0032
-2.9813	-.1836	-.0663	4.3762	12.142145293.	.5606	.4328	40.0023
113644 606	8.5460	1.0885	418.9813	-4.2763	.7607	0.0000	.0010
-2.7677	-.1803	-.0659	4.3756	12.074945293.	.5606	.4328	40.0023
113644 626	8.6433	.9556	418.9563	-4.2764	.7662	0.0000	-.0027
-2.5648	-.1772	-.0659	4.3745	11.995645293.	.5606	.4328	40.0023
113644 646	8.6723	.8115	419.0046	-4.2769	.7820	0.0000	-.0027
-2.3774	-.1745	-.0654	4.3738	11.910545293.	.6716	.4330	40.0450
113644 666	8.6786	.6673	419.0882	-4.2769	.7939	0.0000	-.0013
-2.2070	-.1696	-.0642	4.3764	11.843345293.	.6716	.4330	40.0450
113644 686	8.7033	.5009	419.1549	-4.2769	.7984	0.0000	-.0012
-2.0630	-.1655	-.0628	4.3806	11.764245293.	.6716	.4330	40.0450
113644 706	8.7329	.3383	419.1670	-4.2769	.7943	0.0000	-.0004
-1.9418	-.1622	-.0618	4.3834	11.679145293.	.6716	.4330	40.0450
113644 726	8.7419	.0941	419.1401	-4.2769	.7949	0.0000	.0039
-1.8357	-.1587	-.0602	4.3815	11.612045275.	.0091	.4328	40.0466
113644 746	8.7539	-.1195	419.0932	-4.2769	.8018	0.0000	.0022
-1.7451	-.1535	-.0588	4.3795	11.532945275.	.0091	.4328	40.0466

113644 766	8.7739	-.3505	419.0625	-4.2769	.8156	0.0000	-.0037
-1.6864	-.1473	-.0566	4.3789	11.447945293.6719		.4333	40.0450
113644 786	8.7786	-.5907	419.0360	-4.2769	.8190	0.0000	-.0048
-1.6607	-.1423	-.0556	4.3829	11.380945275.0091		.4328	40.0466
113644 806	8.7502	-.8400	419.0184	-4.2769	.7945	0.0000	.0018
-1.6615	-.1348	-.0537	4.3889	11.301945275.0091		.4328	40.0466
113644 826	8.7298	-1.0738	419.0535	-4.2769	.7743	0.0000	.0043
-1.6807	-.1291	-.0529	4.3930	11.216945275.1194		.4333	40.0892
113644 846	8.7272	-1.2934	419.1332	-4.2828	.7873	0.0000	.0004
-1.7187	-.1266	-.0527	4.3914	11.170845275.1194		.4330	40.0892
113644 866	8.7146	-1.5205	419.2000	-4.3396	.8152	0.0000	-.0033
-1.7743	-.1277	-.0536	4.3882	11.111645275.1194		.4330	40.0892
113644 886	8.6893	-1.7418	419.2166	-4.4277	.8100	0.0000	-.0031
-1.8560	-.1251	-.0533	4.3884	11.027145275.1194		.4330	40.0892
113644 906	8.6742	-1.9330	419.2162	-4.4984	.7912	0.0000	-.0020
-1.9738	-.1215	-.0524	4.3915	10.932245275.1194		.4330	40.0892
113644 926	8.6656	-2.1052	419.2213	-4.5159	.7940	0.0000	-.0026
-2.1355	-.1170	-.0511	4.3958	10.840445275.1194		.4330	40.0892
113644 946	8.6498	-2.2618	419.3004	-4.5155	.8016	0.0000	-.0069
-2.3129	-.1125	-.0492	4.3980	11.755145275.2297		.4333	40.1318
113644 966	8.6378	-2.4013	419.4558	-4.5424	.7913	0.0000	-.0081
-2.4590	-.1082	-.0473	4.3973	10.688645275.3409		.4335	40.1744
113644 986	8.6115	-2.5271	419.6267	-4.6196	.7717	0.0000	-.0051
-2.5464	-.1035	-.0458	4.3960	10.609745275.3409		.4335	40.1744
113645 6	8.5581	-2.6443	419.7122	-4.7082	.7707	0.0000	-.0020
-2.5943	-.0977	-.0441	4.3963	10.525045256.8047		.4335	40.2186
113645 26	8.5013	-2.7288	419.7328	-4.7500	.7729	0.0000	-.0035
-2.5975	-.0916	-.0425	4.3983	10.458145256.8047		.4335	40.2186
113645 46	8.4688	-2.7983	419.7175	-4.7817	.7633	0.0000	-.0047
-2.6131	-.0874	-.0414	4.4000	10.379345256.8047		.4335	40.2186
113645 66	8.4600	-2.8753	419.7082	-4.8579	.7527	0.0000	-.0048
-2.6390	-.0854	-.0413	4.4010	10.294745256.8047		.4335	40.2186
113645 86	8.4477	-2.9584	419.7077	-4.9464	.7490	0.0000	-.0074
-2.6769	-.0821	-.0411	4.4019	10.234645256.8047		.4335	40.2186
113645 106	8.4071	-3.0469	419.7079	-4.9882	.7458	0.0000	-.0074
-2.7171	-.0784	-.0385	4.4029	10.174645256.8047		.4335	40.2186
113645 126	8.3293	-3.1199	419.7079	-5.0199	.7457	0.0000	-.0035
-2.7486	-.0745	-.0367	4.4017	10.105845256.8047		.4335	40.2186
113645 146	8.2445	-3.1695	419.7079	-5.0962	.7499	0.0000	-.0025
-2.7736	-.0726	-.0359	4.4001	10.020545256.8047		.4335	40.2186
113645 166	8.1760	-3.2038	419.7336	-5.1847	.7442	0.0000	-.0035
-2.8039	-.0699	-.0351	4.3995	9.931745256.8047		.4335	40.2186
113645 186	8.1184	-3.2466	419.8111	-5.2266	.7282	0.0000	-.0017
-2.8266	-.0665	-.0340	4.3992	9.858145256.9148		.4338	40.2612
113645 206	8.0649	-3.2727	419.9086	-5.2583	.7167	0.0000	.0020
-2.8439	-.0625	-.0323	4.3997	9.786045238.3959		.4338	40.3054
113645 226	8.0137	-3.2773	419.9643	-5.3345	.7133	0.0000	.0023
-2.8853	-.0588	-.0308	4.4023	9.700745238.3959		.4338	40.3054
113645 246	7.9761	-3.2797	420.0017	-5.4231	.7125	0.0000	-.0008
-2.9498	-.0526	-.0281	4.4052	9.627045238.3959		.4338	40.3054
113645 266	7.9426	-3.2937	420.0738	-5.4650	.7159	0.0000	-.0052
-3.0133	-.0461	-.0252	4.4063	9.554845238.5059		.4341	40.3480
113645 286	7.9113	-3.3031	420.1619	-5.4967	.7129	0.0000	-.0057
-3.0689	-.0404	-.0225	4.4067	9.469645238.5059		.4341	40.3480
113645 306	7.8866	-3.2849	420.2542	-5.5730	.7030	0.0000	-.0023
-3.1227	-.0377	-.0212	4.4065	9.375045238.6159		.4343	40.3906
113645 326	7.8637	-3.2346	420.3407	-5.6616	.6975	0.0000	-.0011
-3.1774	-.0358	-.0204	4.4072	9.283145238.6159		.4343	40.3906
113645 346	7.8343	-3.1853	420.4059	-5.7094	.6987	0.0000	-.0024
-3.2424	-.0336	-.0195	4.4096	9.197645238.6159		.4343	40.3906
113645 366	7.7938	-3.1267	420.4222	-5.7706	.6958	0.0000	-.0021
-3.3163	-.0304	-.0182	4.4147	9.131045238.6159		.4343	40.3906
113645 386	7.7609	-3.0526	420.4067	-5.8579	.6948	0.0000	-.0011
-3.3890	-.0248	-.0156	4.4204	9.052145238.6159		.4343	40.3906
113645 406	7.7386	-2.9684	420.4014	-5.9286	.6909	0.0000	-.0010
-3.4530	-.0218	-.0144	4.4238	8.967245220.1132		.4343	40.4348

113645 426	7.7170	-2.8894	420.4174	-5.9521	.6837	0.0000	-.0003
-3.5058	-.0193	-.0132	4.4247	8.900345220	.1132	.4343	40.4348
113645 446	7.6816	-2.8094	420.4575	-6.0086	.6786	0.0000	.0005
-3.5504	-.0159	-.0116	4.4247	8.821445220	.1132	.4343	40.4348
113645 465	7.6338	-2.7182	420.5238	-6.0965	.6781	0.0000	-.0011
-3.5956	-.0119	-.0096	4.4256	8.710045220	.2231	.4346	40.4774
113645 486	7.5847	-2.6226	420.6065	-6.1673	.6744	0.0000	-.0020
-3.6555	-.0082	-.0078	4.4294	8.595045220	.2231	.4346	40.4774
113645 506	7.5552	-2.5170	420.6721	-6.1849	.6695	0.0000	.0001
-3.7323	-.0035	-.0051	4.4342	8.505145220	.2231	.4346	40.4774
113645 526	7.5313	-2.3999	420.6885	-6.2120	.6708	0.0000	.0017
-3.8180	.0036	-.0006	4.4390	8.439145220	.2231	.4346	40.4774
113645 546	7.5184	-2.2611	420.7137	-6.2889	.6743	0.0000	-.0021
-3.8959	.0089	.0028	4.4432	8.360545220	.2231	.4346	40.4774
113645 566	7.5063	-2.1321	420.7853	-6.3777	.6745	0.0000	-.0054
-3.9672	.0105	.0040	4.4473	8.249245220	.3330	.4348	40.5200
113645 586	7.4939	-2.0011	420.8474	-6.4197	.6675	0.0000	-.0043
-4.0226	.0120	.0052	4.4513	8.134345220	.3330	.4348	40.5200
113645 606	7.4746	-1.8571	420.8540	-6.4240	.6639	0.0000	-.0018
-4.0628	.0132	.0063	4.4554	8.038945201	.7367	.4346	40.5216
113645 626	7.4676	-1.6864	420.8891	-6.4229	.6641	0.0000	-.0019
-4.1005	.0133	.0065	4.4590	7.944645201	.8465	.4348	40.5642
113645 646	7.4631	-1.5314	421.0368	-6.4228	.6689	0.0000	-.0020
-4.1422	.0137	.0071	4.4609	7.849745201	.9562	.4351	40.6067
113645 666	7.4456	-1.3804	421.2269	-6.4229	.6671	0.0000	-.0016
-4.1955	.0166	.0097	4.4624	7.727545202	.0660	.4353	40.6493
113645 686	7.4413	-1.2226	421.3786	-6.4504	.6648	0.0000	-.0016
-4.2564	.0203	.0131	4.4638	7.619945202	.0660	.4353	40.6493
113645 706	7.4629	-1.0473	421.4925	-6.5278	.6626	0.0000	-.0019
-4.3117	.0219	.0142	4.4656	7.497245202	.1757	.4355	40.6919
113645 726	7.4745	-.8675	421.5833	-6.6167	.6620	0.0000	-.0017
-4.3499	.0210	.0146	4.4671	7.369245202	.1757	.4355	40.6919
113645 746	7.4770	-.6992	421.6535	-6.6586	.6590	0.0000	-.0015
-4.3872	.0222	.0164	4.4683	7.227045202	.2854	.4358	40.7344
113645 766	7.4991	-.5241	421.6976	-6.6629	.6563	0.0000	-.0029
-4.4188	.0225	.0171	4.4700	7.119545183	.7053	.4356	40.7361
113645 786	7.5413	-.3301	421.7111	-6.6619	.6609	0.0000	-.0043
-4.4426	.0224	.0174	4.4732	7.024845183	.7053	.4356	40.7361
113645 806	7.5611	-.1359	421.6962	-6.6618	.6728	0.0000	-.0035
-4.4621	.0233	.0188	4.4747	6.969745183	.7053	.4356	40.7361
113645 826	7.5715	.0371	421.6965	-6.6618	.6780	0.0000	.0000
-4.4860	.0267	.0230	4.4734	6.768245183	.7053	.4356	40.7361
113645 846	7.5877	.1976	421.7585	-6.6618	.6776	0.0000	.0016
-4.5114	.0293	.0266	4.4698	6.639645183	.8149	.4358	40.7786

OUTPUT LISTING

```
                    AIRCRAFT A CHECK CASE                                                    09/17/74
                                        NEWTON-RAPHSON DIGITAL DERIVATIVE MATCHING
                                                      1 APR 1974

INPUT DATA (T INDICATES TRUE OR YES, F INDICATES FALSE OR NO)

    LATERAL      CASE
    DATA SOURCE    CARD? T        TAPE? F
    DATA RATE IS   0. SAMPLES/SECOND ON SOURCE FILE (IF 0, DETERMINED FROM TIMES ON THE SOURCE FILE)
        DIVIDED BY THINNING FACTOR OF  1
    ON INPUT TAPE:  15 DATA WORDS PER RECORD.   SPECIAL SIGNAL ORDER DEFAULT? F


PROGRAM OPTIONS

    APRIORI WEIGHTING =0.                    0 TIME HALVINGS IN EAT.
    ITERATIONS =  6  (ITERATION WILL STOP IF ERROR SUM CHANGES BY LESS THAN A FACTOR OF  .10E-02)
    CASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN  .10E+21


OUTPUT

    PLOTS? T  (NO PLOTS UNLESS FINAL ERROR SUM IS LESS THAN .100E+06)
         NUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED =  8
         SECONDS PER CENTIMETER = 1.00
    PRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES? F
    EXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? F
    PUNCHED FINAL NON-DIMENSIONAL DERIVATIVES AND CONFIDENCE LEVELS? F
    PUNCHED FINAL DIMENSIONAL MATRICES? F


FLIGHT CONDITION AND VEHICLE CHACTERISTICS (0. INDICATES VALUE OBTAINED FROM TIME HISTORY ON QBAR,V OR MACH)
                                        (MACH,ALPHA,CG AND PARAM ARE FOR REFERENCE ONLY, NOT USED IN PROGRAM)
    METRIC UNITS? F
    DYNAMIC PRESSURE =       520.0        VELOCITY = 4665.0
    MACH = 0.000                          ALPHA = 999.00  (IF 999. , OBTAINED FROM TIME HISTORY)
    CENTER OF GRAVITY =   .250            OTHER IDENTIFYING PARAMETER = 0.
    WING AREA =      .0                   SPAN =     .00     CHORD =    .00
    IX =*********                         IY =**********    IZ =**********.   IXZ =      0.0
    WEIGHT =*********
    INSTRUMENT OFFSETS FROM CG
         X-DIRECTION OFFSETS (+ = INSTRUMENT IS FORWARD OF CG)
             ALPHA    0.000  AN    0.000
             BETA     0.000  AY    0.000
         Z-DIRECTION OFFSETS ( + = INSTRUMENT IS BELOW CG)
             BETA     0.000  AY    0.000

SIGNAL SCALING AND BIASES
 SIGNALS       BETA     P        R        PHI      AY       PDOT     RDOT     DA       DR       DC1      DC2      ALFA     V        MACH     QBAR
 VAR BIAS                                          T        T        T
 VAR I.C.       F        F        F        F
 FIXED BIAS    0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
 SCALE FACT    1.00     1.00     1.00     1.00     1.00     1.00     1.00
 PLOT LIMITS
    MINIMUM    0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
    MAXIMUM    0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00


 MANEUVER   1  START TIME    0    0    0    0  STOP TIME    0    0    5  875
```

```
                         AIRCRAFT A CHECK CASE

 INPUT MATRICES :
A                                          4  BY  4
           -.3800E-01  .1110E+00  -.1000E+01  .6900E-02
           -.1679E+02 -.2410E+00   .4000E+00  0.
            .1550E+01 -.2840E-02  -.4200E-01  0.
           -0.         .1000E+01 -0.         -0.
B                                          4  BY  5
           -0.         .1480E-01 -0.         -0.         -0.
            .1276E+02  .2008E+02 -0.         -0.         -0.
            .3577E+00 -.2445E+01 -0.         -0.         -0.
           -0.        -0.        -0.         -0.         -0.
D1                                         7  BY  7
            .2160E+04  0.         0.          0.          0.          0.          0.
            0.         .6500E+01  0.          0.          0.          0.          0.
            0.         0.         .4860E+04   0.          0.          0.          0.
            0.         0.         0.          .1350E+03   0.          0.          0.
            0.         0.         0.          0.          .2265E+02   0.          0.
            0.         0.         0.          0.          0.          .2700E+01   0.
            0.         0.         0.          0.          0.          0.          .1980E+03
                              TOTAL NUMBER OF POINTS FOR MANEUVER  1 =   235
```

```
                              AIRCRAFT A CHECK CASE                                                    09/17/74

          STARTING VALUES      MACH = 0.000      ALPHA =   0.00      PARAM =     0.0000      CG =  .250

DIMENSIONAL DERIVATIVES / SEC / SEC**2
          BETA              P                 R                 DA                DR              DC1             DC2             DELTA-0
 Y      -.038000          .111000        -1.000000*        -0.000000          .014800       -0.000000*      -0.000000*      -0.000000
 L    -16.790000         -.241000          .400000         12.760000        20.080000       -0.000000*      -0.000000*      -0.000000
 N      1.550000         -.002840         -.042000          .357700        -2.445000       -0.000000*      -0.000000*      -0.000000

NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY /RAD)
          BETA              P                 R                 DA                DR              DC1             DC2             DELTA-0
CY**************        0.000000        0.000000*         0.000000  **************        0.000000*       0.000000*       0.000000
CL**************  **************  **************  **************  **************        0.000000*       0.000000*       0.000000
CN**************  **************  **************  **************  **************        0.000000*       0.000000*       0.000000

                                        ( * ) INDICATES DERIVATIVE HELD FIXED DURING MATCH          NUMBER OF UNKNOWNS = 21

                    ENTERING ITERATION LOOP

 DIMENSIONAL DERIVATIVE MATRICES PER RADIAN. BIASES IN RADIANS.

A                                        4  BY  4
             -.3800E-01    .1110E+00  -.1000E+01    .6900E-02
             -.1679E+02   -.2410E+00   .4000E+00   0.
              .1550E+01   -.2840E-02  -.4200E-01   0.
             -0.           .1000E+01  -0.          -0.
B                                        4  BY  5
             -0.           .1480E-01  -0.          -0.          -0.
              .1276E+02    .2008E+02  -0.          -0.          -0.
              .3577E+00   -.2445E+01  -0.          -0.          -0.
             -0.          -0.         -0.          -0.          -0.
                 AY            PDOT          RDOT
VARIABLE BIAS    0.            0.            0.

                                                                          WEIGHTED ERROR SUM =    .6853E+01

ERRORS
   .1051E-03   .1905E-01   .1562E-03   .4192E-01   .9587E-03   .1459E-01   .1087E-03
WEIGHTED ERRORS
   .2271E+00   .1238E+00   .7593E+00   .5660E+01   .2172E-01   .3940E-01   .2153E-01

                                                 ITERATION NUMBER   1 COMPLETED

A                                        4  BY  4
             -.4692E-01    .1023E+00  -.1000E+01    .6900E-02
             -.2424E+02   -.1403E+00   .2220E+01   0.
              .1289E+01   -.3239E-03  -.1284E+00   0.
             -0.           .1000E+01  -0.          -0.
B                                        4  BY  5
              .2250E-02    .1535E-01  -0.          -0.          -.3084E-02
              .1430E+02    .1740E+02  -0.          -0.           .4124E+00
              .4389E+00   -.2173E+01  -0.          -0.          -.6923E-02
             -0.          -0.         -0.          -0.          -.8447E-02
                 AY            PDOT          RDOT
VARIABLE BIAS    .5277E+00   -.2469E-01   -.4478E-02

                                                                          WEIGHTED ERROR SUM =    .3322E+00

ERRORS
   .4787E-05   .1302E-02   .7344E-05   .1885E-02   .3279E-03   .3593E-02   .3082E-04
WEIGHTED ERRORS
   .1034E-01   .8462E-02   .3569E-01   .2545E+00   .7428E-02   .9701E-02   .6103E-02

                                                 ITERATION NUMBER   2 COMPLETED
```

```
A                                    4  BY  4
              -.4698E-01    .1019E+00  -.1000E+01    .6900E-02
              -.2446E+02   -.1014E+00    .2438E+01   0.
               .1295E+01    .6551E-03  -.1454E+00   0.
             -0.            .1000E+01  -0.          -0.
B                                    4  BY  5
               .2786E-02    .1586E-01  -0.          -0.          -.3038E-02
               .1453E+02    .1785E+02  -0.          -0.           .4110E+00
               .4927E+00   -.2133E+01  -0.          -0.          -.7518E-02
             -0.          -0.          -0.          -0.          -.7921E-02
                 AY           PDOT         RDOT
VARIABLE BIAS     .5193E+00   -.2585E-01   -.4465E-02

ERRORS                                                                          WEIGHTED ERROR SUM =    .5911E-01
   .4609E-05    .8052E-03    .2007E-05    .6540E-04    .3544E-03    .4336E-02    .2834E-04
WEIGHTED ERRORS
   .9956E-02    .5234E-02    .9752E-02    .8828E-02    .8027E-02    .1171E-01    .5611E-02


                                                ITERATION NUMBER   3 COMPLETED
A                                    4  BY  4
              -.4671E-01    .1026E+00  -.1000E+01    .6900E-02
              -.2432E+02   -.1018E+00    .2456E+01   0.
               .1290E+01    .4784E-03  -.1509E+00   0.
             -0.            .1000E+01  -0.          -0.
B                                    4  BY  5
               .2732E-02    .1593E-01  -0.          -0.          -.3034E-02
               .1447E+02    .1787E+02  -0.          -0.           .4090E+00
               .5052E+00   -.2126E+01  -0.          -0.          -.7531E-02
             -0.          -0.          -0.          -0.          -.8406E-02
                 AY           PDOT         RDOT
VARIABLE BIAS     .5178E+00   -.2581E-01   -.4472E-02

ERRORS                                                                          WEIGHTED ERROR SUM =    .5890E-01
   .4491E-05    .8141E-03    .2089E-05    .6286E-04    .3535E-03    .4314E-02    .2837E-04
WEIGHTED ERRORS
   .9701E-02    .5292E-02    .1015E-01    .8487E-02    .8006E-02    .1165E-01    .5617E-02


                                                ITERATION NUMBER   4 COMPLETED
A                                    4  BY  4
              -.4670E-01    .1026E+00  -.1000E+01    .6900E-02
              -.2432E+02   -.1015E+00    .2463E+01   0.
               .1290E+01    .4523E-03  -.1514E+00   0.
             -0.            .1000E+01  -0.          -0.
B                                    4  BY  5
               .2752E-02    .1594E-01  -0.          -0.          -.3035E-02
               .1447E+02    .1787E+02  -0.          -0.           .4092E+00
               .5060E+00   -.2125E+01  -0.          -0.          -.7550E-02
             -0.          -0.          -0.          -0.          -.8423E-02
                 AY           PDOT         RDOT
VARIABLE BIAS     .5180E+00   -.2582E-01   -.4472E-02

ERRORS                                                                          WEIGHTED ERROR SUM =    .5890E-01
   .4494E-05    .8136E-03    .2085E-05    .6275E-04    .3539E-03    .4321E-02    .2835E-04
WEIGHTED ERRORS
   .9708E-02    .5288E-02    .1014E-01    .8472E-02    .8017E-02    .1167E-01    .5614E-02


                                                ITERATION NUMBER   5 COMPLETED
```

```
 ITERATION TERMINATING, ERROR WITHIN  .001000  BOUND.

CONFIDENCE LEVELS FOR NEXT TO LAST ITERATION
    (DIMENSIONAL)

AC                                    3  BY  3
              .5401E-03   .7999E-03  0.
              .1508E+00   .9145E-02    .2088E+00
              .1056E-01   .7281E-03    .1637E-01
BC                                    3  BY  5
              .9128E-03   .1019E-02  0.          0.          .2101E-03
              .3688E+00   .3906E+00  0.          0.          .7108E-02
              .3197E-01   .3726E-01  0.          0.          .5759E-03

    (NON-DIMENSIONAL)
AC                                    3  BY  3
              .2629E+07   .7999E-03  0.
              .5063E+10   .1641E+18    .3746E+19
              .3543E+09   .1306E+17    .2938E+18
BC                                    3  BY  5
              .4442E+07   .4959E+07  0.          0.          .5859E+08
              .1238E+11   .1311E+11  0.          0.          .1367E+11
              .1073E+10   .1251E+10  0.          0.          .1107E+10
```

AIRCRAFT A CHECK CASE 09/17/74

FINAL VALUES MACH = 0.000 ALPHA = 0.00 PARAM = 0.0000 CG = .250

DIMENSIONAL DERIVATIVES / SEC / SEC**2

	BETA	P	R	DA	DR	DC1	DC2	DELTA-0
Y	-.046703	.102595	-1.000000*	.002753	.015943	-0.000000*	-0.000000*	-.003035
L	-24.320923	-.101510	2.463745	14.469799	17.868306	-0.000000*	-0.000000*	.409219
N	1.289852	.000448	-.151448	.506167	-2.124599	-0.000000*	-0.000000*	-.007553

NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY /RAD)

	BETA	P	R	DA	DR	DC1	DC2	DELTA-0
CY**********		0.000000	0.000000*	**********	**********	0.000000*	0.000000*	**********
CL**********	**********	**********	**********	**********	**********	0.000000*	0.000000*	**********
CN**********	**********	**********	**********	**********	**********	0.000000*	0.000000*	**********

(*) INDICATES DERIVATIVE HELD FIXED DURING MATCH

VARIABLE BIAS .5179E+00 -.2582E-01 -.4472E-02

FINAL DIMENSIONAL MATRICES

A 4 BY 4

-.4670E-01	.1026E+00	-.1000E+01	.6900E-02
-.2432E+02	-.1015E+00	.2464E+01	0.
.1290E+01	.4483E-03	-.1514E+00	0.
-0.	.1000E+01	-0.	-0.

B 4 BY 5

.2753E-02	.1594E-01	-0.	-0.	-.3035E-02
.1447E+02	.1787E+02	-0.	-0.	.4092E+00
.5062E+00	-.2125E+01	-0.	-0.	-.7553E-02
-0.	-0.	-0.	-0.	-.8423E-02

DEGREES	AY	PDOT	RDOT
VARIABLE BIAS	.5179E+00	-.1479E+01	-.2562E+00

WEIGHTED ERROR SUM = .5890E-01

ERRORS

.4494E-05	.8135E-03	.2085E-05	.6274E-04	.3540E-03	.4322E-02	.2835E-04

WEIGHTED ERRORS

.9707E-02	.5287E-02	.1014E-01	.8471E-02	.8017E-02	.1167E-01	.5613E-02

ERRORS

6.85	.33	.06	.06	.06	.06

```
                    AIRCRAFT B CHECK CASE                                                    09/17/74
                                    NEWTON-RAPHSON DIGITAL DERIVATIVE MATCHING
                                                    1 APR 1974

INPUT DATA (T INDICATES TRUE OR YES, F INDICATES FALSE OR NO)

    LONGITUDINAL CASE
    DATA SOURCE    CARD? T         TAPE? F
    DATA RATE IS  50. SAMPLES/SECOND ON SOURCE FILE (IF 0, DETERMINED FROM TIMES ON THE SOURCE FILE)
         DIVIDED BY THINNING FACTOR OF  1
    ON INPUT TAPE:  25 DATA WORDS PER RECORD.   SPECIAL SIGNAL ORDER DEFAULT? T


PROGRAM OPTIONS

    APRIORI WEIGHTING = .10E+01            0 TIME HALVINGS IN EAT.
    ITERATIONS =  6  (ITERATION WILL STOP IF ERROR SUM CHANGES BY LESS THAN A FACTOR OF  .10E-02)
    CASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN  .10E+21


OUTPUT

    PLOTS? T  (NO PLOTS UNLESS FINAL ERROR SUM IS LESS THAN .100E+06)
         NUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED =  8
         SECONDS PER CENTIMETER =  .50
    PRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES? F
    EXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? F
    PUNCHED FINAL NON-DIMENSIONAL DERIVATIVES AND CONFIDENCE LEVELS? T
    PUNCHED FINAL DIMENSIONAL MATRICES? F


FLIGHT CONDITION AND VEHICLE CHACTERISTICS (0. INDICATES VALUE OBTAINED FROM TIME HISTORY ON QBAR,V OR MACH)
                                           (MACH,ALPHA,CG AND PARAM ARE FOR REFERENCE ONLY, NOT USED IN PROGRAM)
    METRIC UNITS? F
    DYNAMIC PRESSURE =       39.0       VELOCITY =  415.2
    MACH =  .429                        ALPHA =   7.86  (IF 999. , OBTAINED FROM TIME HISTORY)
    CENTER OF GRAVITY =  .260           OTHER IDENTIFYING PARAMETER =  .500E+01
    WING AREA =   85.0                  SPAN =  16.05     CHORD =  5.98
    IX =    275.0                       IY =    1932.0    IZ =    2228.0    IXZ =    11.6
    WEIGHT =   2470.0
    INSTRUMENT OFFSETS FROM CG
         X-DIRECTION OFFSETS (+ = INSTRUMENT IS FORWARD OF CG)
             ALPHA   0.000  AN    -.010
             BETA    0.000  AY    0.000
         Z-DIRECTION OFFSETS ( + = INSTRUMENT IS BELOW CG)
             BETA    0.000  AY    0.000

SIGNAL SCALING AND BIASES
 SIGNALS       ALFA     Q        V        THET     AN       QDOT     AX       DE       DC       DC1      DC2      PHI      ALT      MACH     QBAR
 VAR BIAS                                          T        T        T
 VAR I.C.       F        F        F        F
 FIXED BIAS    0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
 SCALE FACT    1.00     1.00     1.00     1.00     1.00     1.00     1.00
 PLOT LIMITS
    MINIMUM    0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
    MAXIMUM    0.00     0.00  1000.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00


 MANEUVER   1  START TIME   11   36   38  750  STOP TIME   11   36   45  840
```

```
                    AIRCRAFT B CHECK CASE

 INPUT MATRICES :
A                                      4  BY  4
              -.4204E+00   .1000E+01 -0.          .2210E-02
              -.3794E+01 -.3632E+00  0.          0.
               .1567E+02  0.         -0.         -.3216E+02
              0.           .9916E+00  0.         0.
B                                      4  BY  5
              -.6489E-01 -0.         -0.         -0.          .7346E-01
              -.6281E+01  0.          0.          0.          .1817E+00
               .8354E+01 -0.         -0.         -0.         -.2394E+01
             -0.           .1000E+01 -0.         -0.         -0.
D1                                     5  BY  5
               .1000E+06  0.          0.          0.         0.
              0.           .7000E+05  0.          0.         0.
              0.          0.         -0.          0.         0.
              0.          0.          0.           .4000E+06 0.
              0.          0.          0.          0.          .1000E+05
APRA                                   4  BY  4
               .5000E+05 -0.         -0.         -0.
               .3000E+00  .1000E+04 -0.         -0.
              0.         -0.         -0.         -0.
              0.         -0.         -0.         -0.
APRB                                   4  BY  8
               .1000E+05 -0.         -0.         -0.         -0.         -0.         -0.         -0.
               .1000E+01 -0.         -0.         -0.         -0.         -0.         -0.         -0.
              0.         -0.         -0.         -0.         -0.         -0.         -0.         -0.
              0.         -0.         -0.         -0.         -0.         -0.         -0.         -0.
                                  TOTAL NUMBER OF POINTS FOR MANEUVER  1 =   354
```

```
                              AIRCRAFT B CHECK CASE                                                    09/17/74

          STARTING VALUES     MACH =  .429     ALPHA =   7.86     PARAM =    5.0000     CG =  .260

DIMENSIONAL DERIVATIVES / SEC / SEC**2
          ALFA            Q               V              DE             DC            DC1           DC2          DELTA-0
 N      .420350       -1.000000*       0.000000*       .064890       0.000000*     0.000000*     0.000000*     -.073460
 M    -3.794300        -.363210        0.000000*     -6.280730       0.000000*     0.000000*     0.000000*      .181650
 A   -15.668030*      -0.000000*       0.000000*     -8.353920*      0.000000*     0.000000*     0.000000*     2.393650*

NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY /RAD)
          ALFA            Q               V              DE             DC            DC1           DC2          DELTA-0
CN      .070547        0.000000*       0.000000*       .010890       0.000000*     0.000000*     0.000000*     -.706389
CM     -.006354       -4.839157        0.000000*     -.010518        0.000000*     0.000000*     0.000000*      .017429
CA     -.006333*      -0.000000*       0.000000*     -.003377*       0.000000*     0.000000*     0.000000*      .055437*

                                    ( * ) INDICATES DERIVATIVE HELD FIXED DURING MATCH          NUMBER OF UNKNOWNS =  9

                         ENTERING ITERATION LOOP

 DIMENSIONAL DERIVATIVE MATRICES PER RADIAN. BIASES IN RADIANS.

A                                          4  BY  4
              -.4204E+00    .1000E+01 -0.           .2210E-02
              -.3794E+01   -.3632E+00  0.          0.
               .1567E+02   0.          -0.         -.3216E+02
              0.            .9916E+00   0.          0.
B                                          4  BY  5
              -.6489E-01 -0.           -0.          -0.           .7346E-01
              -.6281E+01  0.            0.          0.            .1817E+00
               .8354E+01 -0.           -0.          -0.          -.2394E+01
             -0.            .1000E+01 -0.          -0.          -0.
                 AN
VARIABLE BIAS     .1000E+01                                                  WEIGHTED ERROR SUM =    .2193E+03

ERRORS
   .1810E-04    .7834E-04    .4079E+02    .4008E-03    .5171E-02
WEIGHTED ERRORS
   .1810E+01    .5484E+01   0.            .1603E+03    .5171E+02


                                                  ITERATION NUMBER   1 COMPLETED
A                                          4  BY  4
              -.4555E+00    .1000E+01 -0.           .2210E-02
              -.3229E+01   -.3851E+00  0.          0.
               .1567E+02   0.          -0.         -.3216E+02
              0.            .9916E+00   0.          0.
B                                          4  BY  5
              -.5102E-01 -0.           -0.          -0.           .8519E-01
              -.6259E+01  0.            0.          0.            .1012E+00
               .8354E+01 -0.           -0.          -0.          -.2394E+01
             -0.            .1000E+01 -0.          -0.           .1491E-02
                 AN
VARIABLE BIAS     .1012E+01                                                  WEIGHTED ERROR SUM =    .6688E+01

ERRORS
   .6808E-05    .1238E-04    .2221E+02    .6785E-05    .2427E-03
WEIGHTED ERRORS
   .6808E+00    .8665E+00   0.            .2714E+01    .2427E+01


                                                  ITERATION NUMBER   2 COMPLETED
```

```
A                                        4  BY  4
              -.4511E+00    .1000E+01 -0.           .2210E-02
              -.3222E+01   -.4853E+00  0.          0.
               .1567E+02   0.          -0.         -.3216E+02
              0.            .9916E+00  0.          0.
B                                        4  BY  5
              -.5149E-01 -0.           -0.         -0.           .8460E-01
              -.6271E+01  0.            0.          0.           .9628E-01
               .8354E+01 -0.           -0.         -0.          -.2394E+01
             -0.            .1000E+01 -0.         -0.           .1631E-02
                 AN
VARIABLE BIAS     .1013E+01

                                                                WEIGHTED ERROR SUM =    .5923E+01
ERRORS
   .5865E-05    .1364E-04    .2252E+02    .6054E-05    .1961E-03
WEIGHTED ERRORS
   .5865E+00    .9547E+00   0.            .2421E+01    .1961E+01


                                                 ITERATION NUMBER   3 COMPLETED
A                                        4  BY  4
              -.4502E+00    .1000E+01 -0.           .2210E-02
              -.3196E+01   -.4956E+00  0.          0.
               .1567E+02   0.          -0.         -.3216E+02
              0.            .9916E+00  0.          0.
B                                        4  BY  5
              -.5192E-01 -0.           -0.         -0.           .8431E-01
              -.6260E+01  0.            0.          0.           .9292E-01
               .8354E+01 -0.           -0.         -0.          -.2394E+01
             -0.            .1000E+01 -0.         -0.           .1526E-02
                 AN
VARIABLE BIAS     .1012E+01

                                                                WEIGHTED ERROR SUM =    .5902E+01
ERRORS
   .5931E-05    .1376E-04    .2251E+02    .6012E-05    .1941E-03
WEIGHTED ERRORS
   .5931E+00    .9632E+00   0.            .2405E+01    .1941E+01


                                                 ITERATION NUMBER   4 COMPLETED
A                                        4  BY  4
              -.4503E+00    .1000E+01 -0.           .2210E-02
              -.3193E+01   -.4999E+00  0.          0.
               .1567E+02   0.          -0.         -.3216E+02
              0.            .9916E+00  0.          0.
B                                        4  BY  5
              -.5195E-01 -0.           -0.         -0.           .8430E-01
              -.6264E+01  0.            0.          0.           .9222E-01
               .8354E+01 -0.           -0.         -0.          -.2394E+01
             -0.            .1000E+01 -0.         -0.           .1521E-02
                 AN
VARIABLE BIAS     .1012E+01

                                                                WEIGHTED ERROR SUM =    .5898E+01
ERRORS
   .5940E-05    .1389E-04    .2251E+02    .5987E-05    .1937E-03
WEIGHTED ERRORS
   .5940E+00    .9722E+00   0.            .2395E+01    .1937E+01


                                                 ITERATION NUMBER   5 COMPLETED

 ITERATION TERMINATING, ERROR WITHIN  .001000  BOUND.
```

```
CONFIDENCE LEVELS FOR NEXT TO LAST ITERATION
    (DIMENSIONAL)

AC                                    3  BY  3
             .4742E-02  0.            0.
             .1515E-01   .1104E-01    0.
            0.          0.            0.
BC                                    3  BY  5
             .1529E-02  0.            0.           0.           .7219E-03
             .4719E-01  0.            0.           0.           .2837E-02
            0.          0.            0.           0.          0.

    (NON-DIMENSIONAL)
AC                                    3  BY  3
             .7959E-03  0.            0.
             .2538E-04   .1471E+00    0.
            0.          0.            0.
BC                                    3  BY  5
             .2566E-03  0.            0.           0.           .6942E-02
             .7902E-04  0.            0.           0.           .2722E-03
            0.          0.            0.           0.          0.
```

AIRCRAFT B CHECK CASE 09/17/74

FINAL VALUES MACH = .429 ALPHA = 7.86 PARAM = 5.0000 CG = .260

DIMENSIONAL DERIVATIVES / SEC / SEC**2

	ALFA	Q	V	DE	DC	DC1	DC2	DELTA-O
N	.450226	-1.000000*	0.000000*	.051967	0.000000*	0.000000*	0.000000*	-.084289
M	-3.192019	-.500318	0.000000*	-6.263662	0.000000*	0.000000*	0.000000*	.092063
A	-15.668030*	-0.000000*	0.000000*	-8.353920*	0.000000*	0.000000*	0.000000*	2.393650*

NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY /RAD)

	ALFA	Q	V	DE	DC	DC1	DC2	DELTA-O
CN	.075562	0.000000*	0.000000*	.008722	0.000000*	0.000000*	0.000000*	-.810524
CM	-.005345	-6.665883	0.000000*	-.010489	0.000000*	0.000000*	0.000000*	.008833
CA	-.006333*	-0.000000*	0.000000*	-.003377*	0.000000*	0.000000*	0.000000*	.055437*

(*) INDICATES DERIVATIVE HELD FIXED DURING MATCH

VARIABLE BIAS .1012E+01

FINAL DIMENSIONAL MATRICES

A 4 BY 4

-.4502E+00	.1000E+01	-0.	.2210E-02
-.3192E+01	-.5003E+00	0.	0.
.1567E+02	0.	-0.	-.3216E+02
0.	.9916E+00	0.	0.

B 4 BY 5

-.5197E-01	-0.	-0.	-0.	.8429E-01
-.6264E+01	0.	0.	0.	.9206E-01
.8354E+01	-0.	-0.	-0.	-.2394E+01
-0.	.1000E+01	-0.	-0.	.1517E-02

DEGREES AN

VARIABLE BIAS .1012E+01

WEIGHTED ERROR SUM = .5898E+01

ERRORS

.5945E-05 .1390E-04 .2251E+02 .5985E-05 .1937E-03

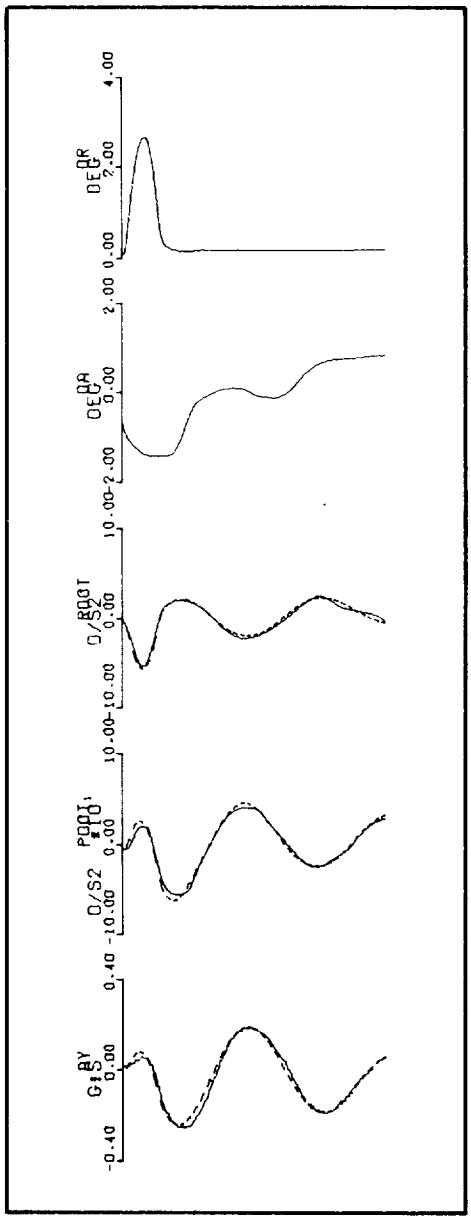
WEIGHTED ERRORS

.5945E+00 .9728E+00 0. .2394E+01 .1937E+01

ERRORS

219.31 6.69 5.92 5.90 5.90 5.90

Two sample plots from the MMLE program are shown. The plots as produced by the automatic plotter are shown within the heavy lines. Explanatory material is included to aid the user in implementing the program. Each plot is presented in two parts to avoid loss of detail from a large reduction. The title on each plot corresponds to the title on the output listing.



δ_r, deg

δ_a, deg

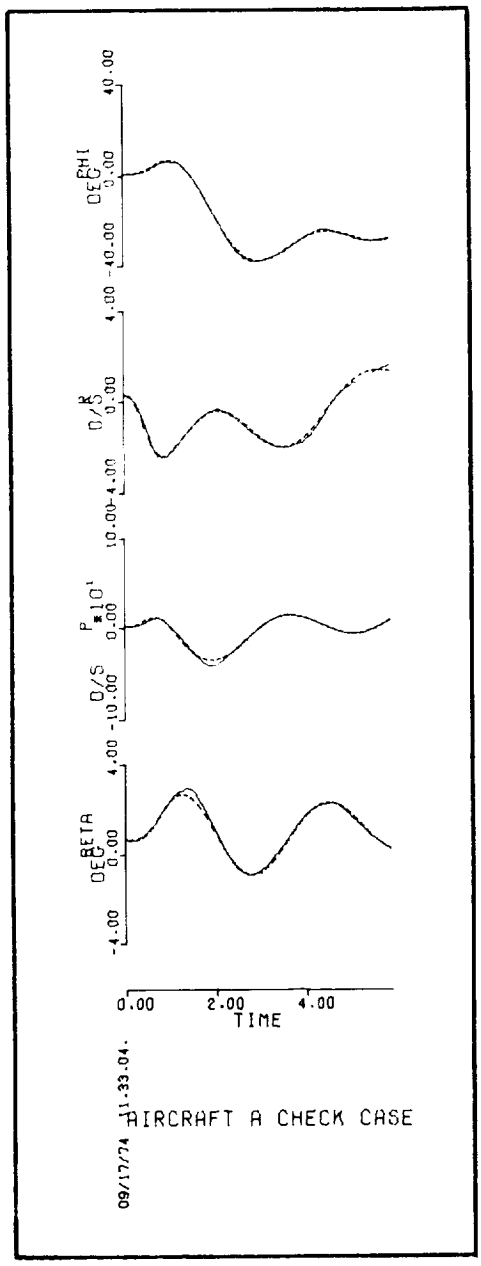
$\dot{r}$, deg/sec^2

$\dot{p}$, deg/sec^2

a_y, g

– – – Computed

——— Flight



– – – Computed

——— Flight

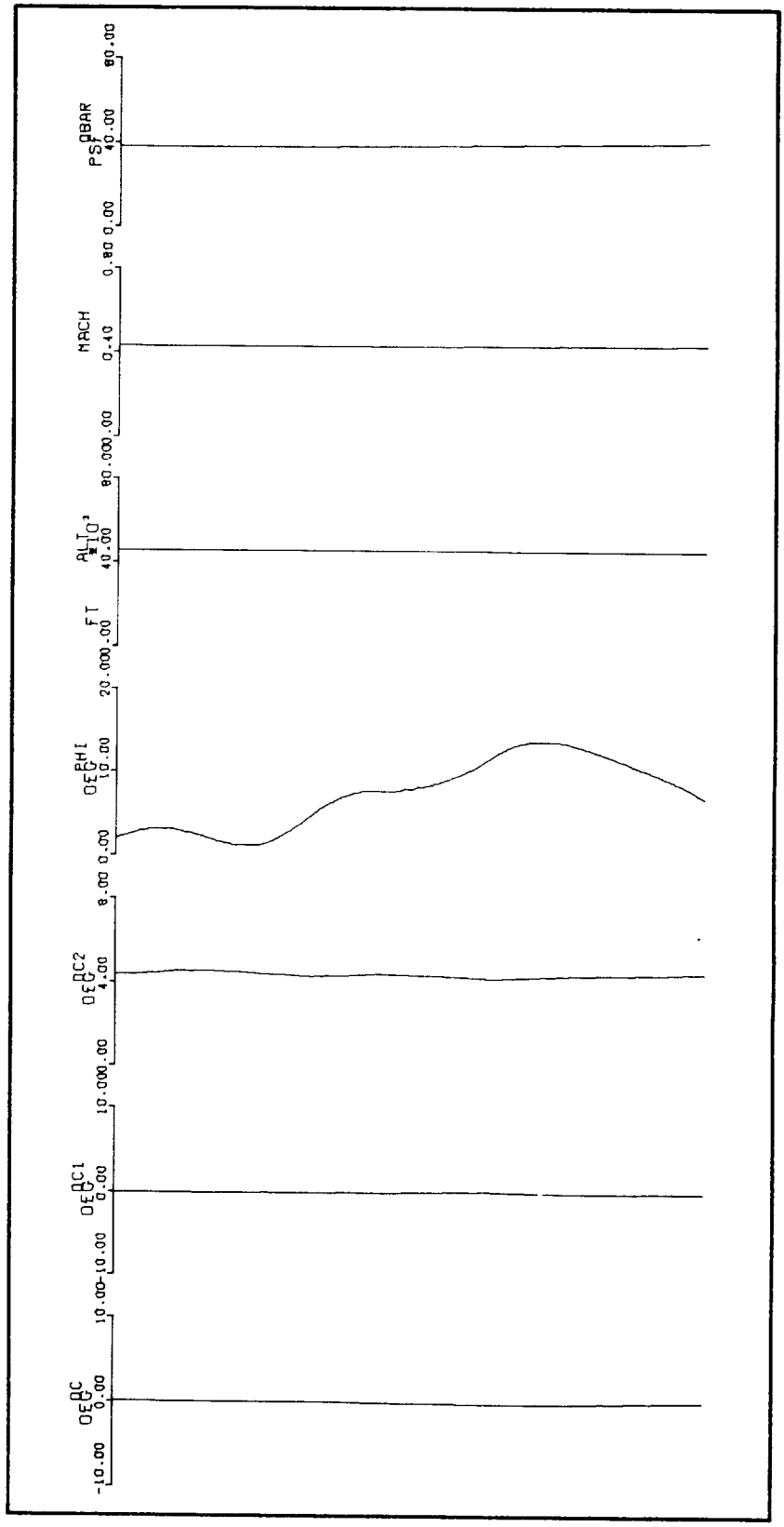
φ, deg

r, deg/sec

p, deg/sec

β, deg

Date and time of run



$\bar{q}$, lb/sq ft

M

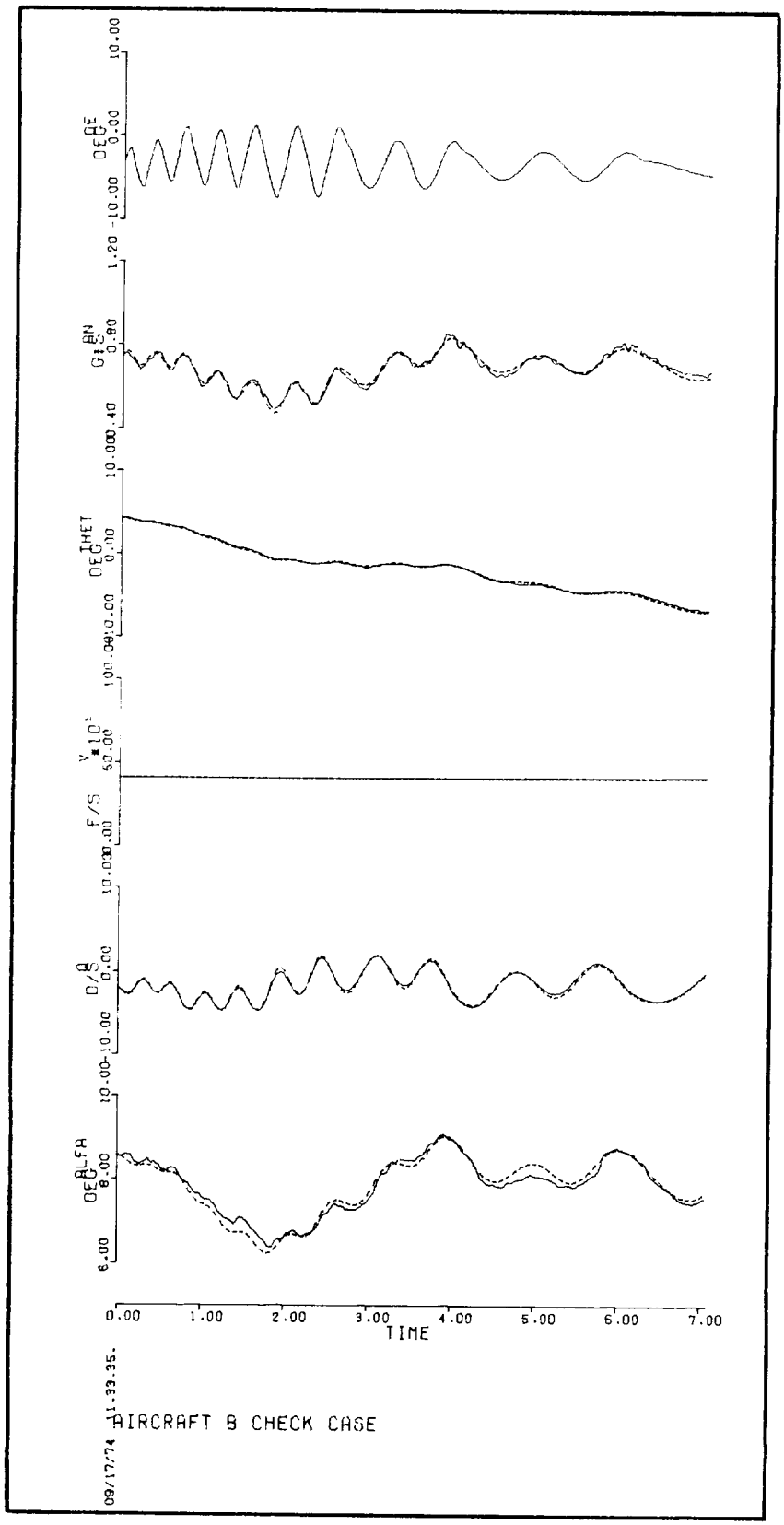
Altitude, ft

φ, deg

δ_{c_2}, deg

δ_{c_1}, deg

δ_c, deg



δ_e, deg

a_n, g

θ, deg

V, ft/sec

q, deg/sec

α, deg

Date and time of run

--- Computed

—— Flight

APPENDIX C

SETUP PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the SETUP program are presented together with supplementary information.

MAIN PROGRAM SETUP

Description: The main SETUP program sets several defaults and then reads the option cards to determine whether it is to read an input tape, punch a card deck, write an output tape, or perform any combination of these operations. It then directs the execution of the assigned tasks for each case.

Programing notes: As in the MMLE program, the program statement is needed on CDC 6000/7000 systems. On an IBM 360/370 system, DD cards perform this function. Cards 590 to 730 are concerned solely with setting the default values for DELTA as defined in the input description (p. 30).

Program listing:

```
      PROGRAM SETUP(INPUT,PUNCH,OUTPUT,TAPE4,TAPE15,TAPE1=INPUT,          MAIN   0
     -     TAPE2=PUNCH,TAPE3=OUTPUT)                                      MAIN  10
      COMMON /ALLDIM/ MAX,MIX                                             MAIN  20
      COMMON /OPTION/ TAPE,DECK,READ                                      MAIN  30
      COMMON /FLCOND/ ALPHA,THETA,Q,V,MACH,IX,IY,IZ,IXZ,W,PHI,CG,KIAS,    MAIN  40
     -     ALT,LONG,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM                 MAIN  50
      REAL MACH,IX,IY,IZ,IXZ,KIAS,AVG(40)                                 MAIN  60
      INTEGER ST(4),ET(4),FLT,CASE                                        MAIN  70
      LOGICAL TAPE,DECK,READ,LONG,LATR,DELTA(4),LATDEL(4),LONDEL(4),DEL   MAIN  80
      NAMELIST /COND/ LONG,LATR,FLT,CASE,ALPHA,THETA,Q,V,MACH,W,IX,IY,    MAIN  90
     -     IZ,IXZ,KIAS,ALT,CG,PARAM,DELTA,PHI,DETRIM                      MAIN 100
      DATA WRT/4HWRIT/,PNC/4HPUNC/,RD/4HREAD/,START/4HSTAR/               MAIN 110
      MAX=5                                                               MAIN 120
      REWIND 15                                                           MAIN 130
      DO 10 I=1,4                                                         MAIN 140
      LONDEL(I)=.FALSE.                                                   MAIN 150
   10 LATDEL(I)=.FALSE.                                                   MAIN 160
      TAPE=.FALSE.                                                        MAIN 170
      DECK=.FALSE.                                                        MAIN 180
      READ=.FALSE.                                                        MAIN 190
      DETRIM=0.                                                           MAIN 200
      PARAM=0.                                                            MAIN 210
      CG=999.                                                             MAIN 220
      IXZ=0.                                                              MAIN 230
      ALT=0.                                                              MAIN 240
      KIAS=0.                                                             MAIN 250
      THETA=0.                                                            MAIN 260
      PHI=0.                                                              MAIN 270
      FLT=0                                                               MAIN 280
      CASE=0                                                              MAIN 290
      LONG=.FALSE.                                                        MAIN 300
C                                                                         MAIN 310
C     READ OPTIONS                                                        MAIN 320
C                                                                         MAIN 330
   20 READ (1,1000) OPTN                                                  MAIN 340
      IF(OPTN.EQ.START) GO TO 50                                          MAIN 350
      IF(OPTN.NE.WRT) GO TO 40                                            MAIN 360
      TAPE=.TRUE.                                                         MAIN 370
      WRITE(3,2001)                                                       MAIN 380
   30 READ=.TRUE.                                                         MAIN 390
      WRITE(3,2000)                                                       MAIN 400
      GO TO 20                                                            MAIN 410
   40 IF(OPTN.EQ.RD) GO TO 30                                             MAIN 420
      IF(OPTN.NE.PNC) GO TO 20                                            MAIN 430
      DECK=.TRUE.                                                         MAIN 440
      WRITE(3,2002)                                                       MAIN 450
      GO TO 20                                                            MAIN 460
   50 IF(DECK) CALL SETIN                                                 MAIN 470
      IF(READ) CALL RDSET                                                 MAIN 480
C                                                                         MAIN 490
C     CASE LOOP                                                           MAIN 500
C                                                                         MAIN 510
  100 READ (1,1001) ST,ET                                                 MAIN 520
      IF(ST(1).LT.0) GO TO 200                                            MAIN 530
      LATR=.FALSE.                                                        MAIN 540
      DO 110 I=1,4                                                        MAIN 550
  110 DELTA(I)=.FALSE.                                                    MAIN 560
```

```
      READ (1,COND)                                                    MAIN 570
      IF(LATR) LONG=.FALSE.                                            MAIN 580
      DEL=DELTA(1).OR.DELTA(2).OR.DELTA(3).OR.DELTA(4)                 MAIN 590
      IF(LONG) GO TO 150                                               MAIN 600
      IF(.NOT.DEL) GO TO 130                                           MAIN 610
      DO 120 I=1,4                                                     MAIN 620
  120 LATDEL(I)=DELTA(I)                                               MAIN 630
      GO TO 190                                                        MAIN 640
  130 DO 140 I=1,4                                                     MAIN 650
  140 DELTA(I)=LATDEL(I)                                               MAIN 660
      GO TO 190                                                        MAIN 670
  150 IF(.NOT.DEL) GO TO 170                                           MAIN 680
      DO 160 I=1,4                                                     MAIN 690
  160 LONDEL(I)=DELTA(I)                                               MAIN 700
      GO TO 190                                                        MAIN 710
  170 DO 180 I=1,4                                                     MAIN 720
  180 DELTA(I)=LONDEL(I)                                               MAIN 730
  190 WRITE(3,2003)FLT,CASE,ST,ET,LONG                                 MAIN 740
      IF(READ) CALL TAPERD                                             MAIN 750
      IF(DECK) CALL PNCH                                               MAIN 760
      GO TO 100                                                        MAIN 770
 1000 FORMAT(A4)                                                       MAIN 780
 1001 FORMAT(2(3I2,I3,1X))                                             MAIN 790
 2000 FORMAT(18HOTAPE WILL BE READ)                                    MAIN 800
 2001 FORMAT(25H0MLE TAPE WILL BE WRITTEN)                             MAIN 810
 2002 FORMAT(25H0MLE DECK WILL BE PUNCHED)                             MAIN 820
 2003 FORMAT(1H1,20X,6HFLIGHT,I3,5X,4HCASE,I4,5X,4HTIME,4I4,4H  TO,    MAIN 830
     -     4I4,5X,14HLONGITUDINAL? ,L1)                                MAIN 840
  200 STOP                                                             MAIN 850
      END                                                              MAIN 860
```

SUBROUTINE SETIN

Description: Subroutine SETIN initializes all information needed to punch the MMLE program deck. It sets several defaults and reads in the values desired. It also calls WINDIN to input predicted derivatives and COND1 to make any other input required by the user.

Subroutine listing:

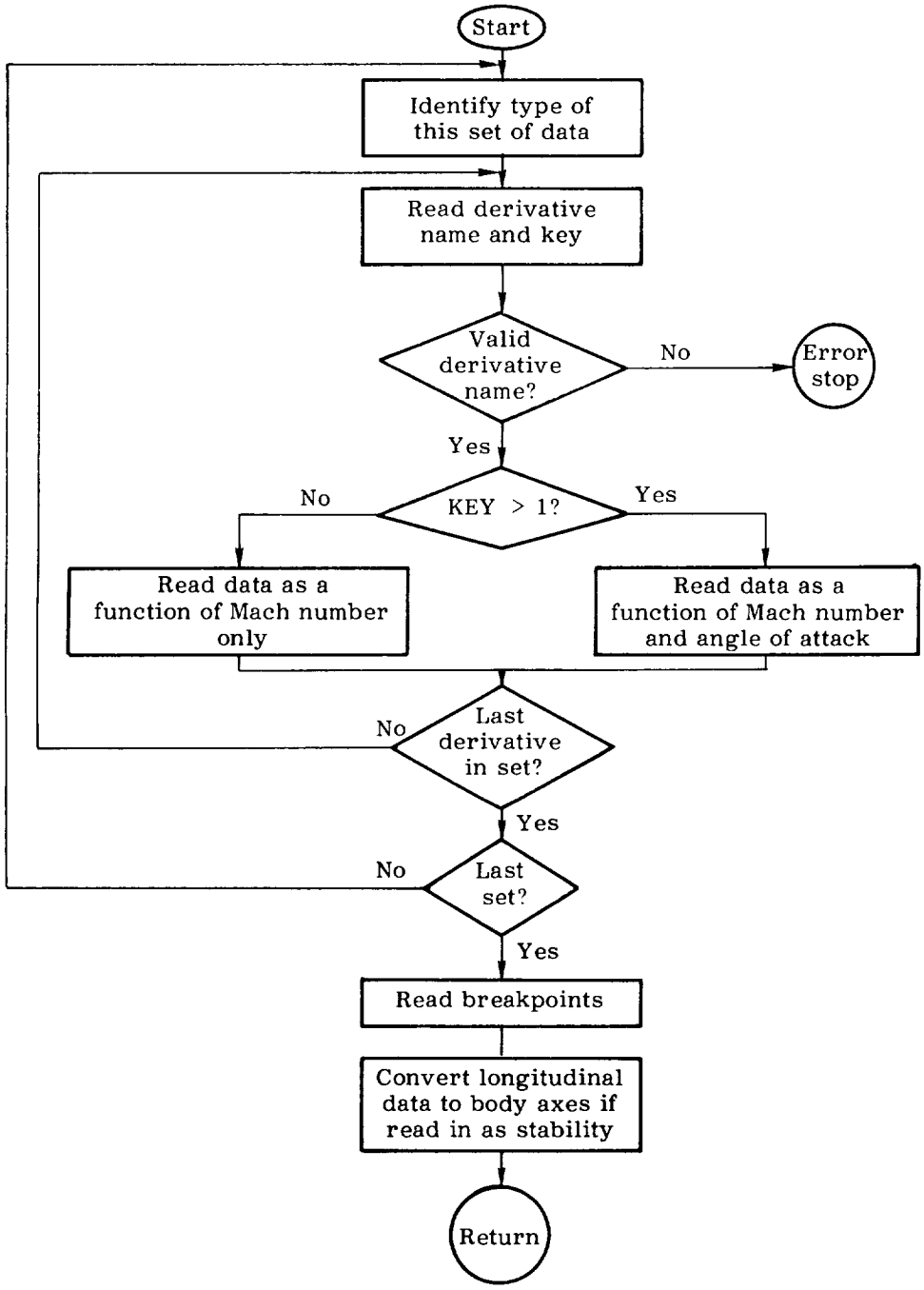
```
      SUBROUTINE SETIN                                                  SETI   0
      COMMON /COM/ MZLA,MZLO,S,SPAN,CBAR,CGLA,CGLC,METRIC,D1LO,         SETI  10
     -     D1LA,VEH,APRALA,APRBLA,APRALO,APRBLO,WMLA,WMLO,PUNCH,CORECT, SETI  20
     -     XALF,XB,ZB,XAN,ZAX,XAY,ZAY,SPS,DLA,DLO                      SETI  30
      COMMON /DATAWT/ NBP,NMBP,NABP,DATA                                SETI  40
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                   SETI  50
      REAL DATA(3000),D1LO(7),D1LA(7),ABP(16),MBP(16),BP(8),VEH(2),     SETI  60
     -     APRALA(5,4),APRALO(5,4),APRBLA(5,8),APRBLO(5,8)             SETI  70
      LOGICAL METRIC,LONG(8),LATR(8),RAD,DEG,PUNCH,CORECT,BODY,STAB,    SETI  80
     -     LAT,LON,DLA,DLO                                             SETI  90
      NAMELIST /WIND/ NABP,NMBP,NBP,S,SPAN,CBAR,CGLA,CGLO,DEG,RAD,      SETI 100
     -     METRIC,LONG,LATR,MZLO,MZLA,NCLO,NCLA,WMLO,WMLA,PUNCH,XALF,   SETI 110
     -     XAN,ZAX,XAY,ZAY,SPS,XB,ZB,BODY,STAB                         SETI 120
C                                                                       SETI 130
C     DEFAULTS                                                          SETI 140
      SPS=0.                                                            SETI 150
      XB=0.                                                             SETI 160
      ZB=0.                                                             SETI 170
      XAY=0.                                                            SETI 180
      ZAY=0.                                                            SETI 190
      XAN=0.                                                            SETI 200
      ZAX=0.                                                            SETI 210
      XALF=0.                                                           SETI 220
      CORECT=.FALSE.                                                    SETI 230
      CGLA=.25                                                          SETI 240
      CGLO=.25                                                          SETI 250
      MZLA=5                                                            SETI 260
      MZLO=5                                                            SETI 270
      PUNCH=.FALSE.                                                     SETI 280
      METRIC=.FALSE.                                                    SETI 290
      BODY=.FALSE.                                                      SETI 300
      DLA=.FALSE.                                                       SETI 310
      DLO=.FALSE.                                                       SETI 320
      WMLA=-99999.                                                      SETI 330
      WMLO=-99999.                                                      SETI 340
      RAD=.FALSE.                                                       SETI 350
      DO 5 I=1,3000                                                     SETI 360
    5 DATA(I)=0.                                                        SETI 370
      DO 10 I=1,8                                                       SETI 380
      BP(I)=0.                                                          SETI 390
      LONG(I)=.TRUE.                                                    SETI 400
   10 LATR(I)=.FALSE.                                                   SETI 410
      NBP=1                                                             SETI 420
      NABP=1                                                            SETI 430
      NMBP=1                                                            SETI 440
      LON=.FALSE.                                                       SETI 450
      LAT=.FALSE.                                                       SETI 460
      READ (1,WIND)                                                     SETI 470
      IF(ABS(XB)+ABS(ZB)+ABS(XALF)+ABS(XAN)+ABS(ZAX)+ABS(XAY)+ABS(ZAY)  SETI 480
     -     .NE. 0.) CORECT=.TRUE.                                       SETI 490
      DO 40 I=1,NBP                                                     SETI 500
      IF(.NOT.LATR(I)) GO TO 30                                         SETI 510
      LONG(I)=.FALSE.                                                   SETI 520
   30 IF(LONG(I)) GO TO 35                                              SETI 530
      LAT=.TRUE.                                                        SETI 540
      GO TO 40                                                          SETI 550
   35 LON=.TRUE.                                                        SETI 560
```

```
  40 CONTINUE                                                          SETI 570
     READ (1,1000) VEH                                                 SETI 580
     IF(LAT) READ (1,1001) D1LA                                        SETI 590
     IF(LON) READ (1,1001) D1LO                                        SETI 600
     IF(D1LA(1)+D1LA(2)+D1LA(3)+D1LA(4)+D1LA(5).GT.0.) DLA=.TRUE.      SETI 610
     IF(D1LO(1)+D1LO(2)+D1LO(3)+D1LO(4)+D1LO(5).GT.0.) DLO=.TRUE.      SETI 620
     IF(WMLA.LT.0.) GO TO 50                                           SETI 630
     CALL LOAD1(APRALA)                                                SETI 640
     CALL LOAD1(APRBLA)                                                SETI 650
  50 IF(WMLO.LT.0.) GO TO 60                                           SETI 660
     CALL LOAD1(APRALO)                                                SETI 670
     CALL LOAD1(APRBLO)                                                SETI 680
  60 WRITE(3,2000)VEH,CGLA,CGLO,RAD                                    SETI 690
     CALL WINDIN(DATA,NBP,NMBP,NABP,BODY,.TRUE.,RAD)                   SETI 700
     CALL COND1                                                        SETI 710
1000 FORMAT(2A4)                                                       SETI 720
1001 FORMAT(7F10.4)                                                    SETI 730
2000 FORMAT(1H1,2A4,5X,27HWIND TUNNEL DATA.  REF CG =,F5.3,7H (LAT),,  SETI 740
    -      F5.3,23H (LONG)     PER RADIAN? ,L1)                       SETI 750
     RETURN                                                            SETI 760
     END                                                               SETI 770
```

SUBROUTINE WINDIN

Description: Subroutine WINDIN reads in predicted derivatives, converting longitudinal data from the stability axes to the body axes if required.

Flow chart:



Programing notes: The loop from cards 770 to 1000 is written in a more expanded form than necessary to improve its efficiency.

Subroutine listing:

```
      SUBROUTINE WINDIN(DATA,NBP,NMBP,NABP,BODY,PRINT,RAD)                  WIND   0
C                                                                           WIND  10
C     READS IN WIND TUNNEL DATA                                             WIND  20
C                                                                           WIND  30
C     DATA IS DIMENSIONED (NBP,NMBP,NABP,NCMAX) WITH THE LAST 2             WIND  40
C     DIMENSIONS HANDLED IN FORTRAN FOR COMPILERS LIMITED TO 3-D            WIND  50
      COMMON /WIDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                       WIND  60
      REAL DATA(NBP,NMBP,1),ABP(16),MBP(16),BP(8),DER(21,3)                 WIND  70
      LOGICAL LONG(8),BODY,PRINT,RAD                                        WIND  80
      INTEGER NNABP(21)                                                     WIND  90
      DATA DER/3HCYB,3HCLB,3HCNB,3HCLP,3HCNP,3HCLR,3HCNR,4HCYDA,4HCLDA,     WIND 100
     -     4HCNDA,4HCYDR,4HCLDR,4HCNDR,4HCYD1,4HCLD1,4HCND1,4HCY02,         WIND 110
     -     4HCLD2,4HCND2,1H  ,1H  ,                                         WIND 120
     -     3HCLA,3HCMA,3HCDA,3HCMQ,3HCLV,3HCMV,3HCDV,4HCLDE,4HCMDE,         WIND 130
     -     4HCDDE,4HCLDC,4HCMDC,4HCDDC,4HCLD1,4HCMD1,4HCOD1,4HCL02,         WIND 140
     -     4HCMD2,4HCOD2,2HCL,2HCD,                                         WIND 150
     -     3HCNA,3HCMA,3HCAA,3HCMQ,3HCNV,3HCMV,3HCAV,4HCNDE,4HCMDE,         WIND 160
     -     4HCADE,4HCNDC,4HCMDC,4HCADC,4HCND1,4HCMD1,4HCAD1,4HCN02,         WIND 170
     -     4HCMD2,4HCAD2,2HCN,2HCA/                                         WIND 180
      NCMAX=21                                                              WIND 190
C     READ NBP SETS OF WIND TUNNEL DATA                                     WIND 200
      DO 200 L=1,NBP                                                        WIND 210
      NC=NCLA                                                               WIND 220
      LL=1                                                                  WIND 230
      IF(.NOT.LONG(L)) GO TO 5                                              WIND 240
      NC=NCLO                                                               WIND 250
      LL=2                                                                  WIND 260
      IF(BODY) LL=3                                                         WIND 270
    5 IF(NC.EQ.0) GO TO 200                                                 WIND 280
      DO 100 II=1,NC                                                        WIND 290
C     READ AND IDENTIFY DERIVATIVE NAME                                     WIND 300
      READ (1,1002) DERIV,N                                                 WIND 310
      IF(PRINT) WRITE(3,2000)DERIV                                          WIND 320
      DO 10 I=1,NCMAX                                                       WIND 330
      IF(DERIV.EQ.DER(I,LL)) GO TO 20                                       WIND 340
   10 CONTINUE                                                              WIND 350
      WRITE(3,2001)DERIV                                                    WIND 360
      STOP                                                                  WIND 370
C     INPUT DATA AS FUNCTION OF MACH AND ALPHA OR MACH ONLY                 WIND 380
   20 K2=I*NABP                                                             WIND 390
      K1=K2-NABP+1                                                          WIND 400
      IF(N.LE.1) GO TO 40                                                   WIND 410
      DO 30 J=1,NMBP                                                        WIND 420
      READ (1,1001) (DATA(L,J,K),K=K1,K2)                                   WIND 430
   30 IF(PRINT) WRITE(3,2002)(DATA(L,J,K),K=K1,K2)                          WIND 440
      GO TO 60                                                              WIND 450
   40 READ (1,1001) (DATA(L,J,K1),J=1,NMBP)                                 WIND 460
      IF(PRINT) WRITE(3,2002)(DATA(L,J,K1),J=1,NMBP)                        WIND 470
      DO 50 J=1,NMBP                                                        WIND 480
      DO 50 K=K1,K2                                                         WIND 490
   50 DATA(L,J,K)=DATA(L,J,K1)                                              WIND 500
   60 IF(.NOT.RAD.OR.(I.GT.3.AND.I.LT.8).OR.I.GT.19) GO TO 100              WIND 510
      DO 70 J=1,NMBP                                                        WIND 520
      DO 70 K=K1,K2                                                         WIND 530
   70 DATA(L,J,K)=DATA(L,J,K)/57.2958                                       WIND 540
  100 CONTINUE                                                              WIND 550
  200 CONTINUE                                                              WIND 560
```

```
C
C     READ BREAKPOINTS                                                  WIND 570
C                                                                       WIND 580
      READ (1,1001) (ABP(J),J=1,NABP)                                   WIND 590
      IF(PRINT) WRITE(3,2003)(ABP(J),J=1,NABP)                          WIND 600
      READ (1,1001)  (MBP(J),J=1,NMBP)                                  WIND 610
      IF(PRINT) WRITE(3,2004)(MBP(J),J=1,NMBP)                          WIND 620
      READ  (1,1001) (BP(J),J=1,8)                                      WIND 630
      IF(PRINT) WRITE(3,2005)(BP(J),J=1,NBP)                            WIND 640
      IF(BODY)  RETURN                                                  WIND 650
C     CONVERT STABILITY TO BODY AXES                                    WIND 660
      DO 210 I=1,21                                                     WIND 670
  210 NNABP(I)=I*NABP                                                   WIND 680
      DO 300 K=1,NABP                                                   WIND 690
      DO 220 I=1,21                                                     WIND 700
  220 NNABP(I)=NNABP(I)+1                                               WIND 710
      SA=SIN(ABP(K)/57.2958)                                            WIND 720
      CA=COS(ABP(K)/57.2958)                                            WIND 730
      DO 300 L=1,NBP                                                    WIND 740
      IF(.NOT.LONG(L)) GO TO 300                                        WIND 750
      DO 280 J=1,NMBP                                                   WIND 760
      TEMP=DATA(L,J,NNABP(19))*CA+DATA(L,J,NNABP(20))*SA                WIND 770
      DATA(L,J,NNABP(20))=DATA(L,J,NNABP(20))*CA-DATA(L,J,NNABP(19))*SA WIND 780
      DATA(L,J,NNABP(19))=TEMP                                          WIND 790
      TEMP=DATA(L,J,K)*CA+DATA(L,J,NNABP(2))*SA+DATA(L,J,NNABP(20))     WIND 800
     -     /57.2958                                                     WIND 810
      DATA(L,J,NNABP(2))=DATA(L,J,NNABP(2))*CA-DATA(L,J,K)*SA-          WIND 820
     -     DATA(L,J,NNABP(19))/57.2958                                  WIND 830
      DATA(L,J,K)=TEMP                                                  WIND 840
      TEMP=DATA(L,J,NNABP( 4))*CA+DATA(L,J,NNABP( 6))*SA                WIND 850
      DATA(L,J,NNABP( 6))=DATA(L,J,NNABP( 6))*CA-DATA(L,J,NNABP( 4))*SA WIND 860
      DATA(L,J,NNABP( 4))=TEMP                                          WIND 870
      TEMP=DATA(L,J,NNABP( 7))*CA+DATA(L,J,NNABP( 9))*SA                WIND 880
      DATA(L,J,NNABP( 9))=DATA(L,J,NNABP( 9))*CA-DATA(L,J,NNABP( 7))*SA WIND 890
      DATA(L,J,NNABP( 7))=TEMP                                          WIND 900
      TEMP=DATA(L,J,NNABP(10))*CA+DATA(L,J,NNABP(12))*SA                WIND 910
      DATA(L,J,NNABP(12))=DATA(L,J,NNABP(12))*CA-DATA(L,J,NNABP(10))*SA WIND 920
      DATA(L,J,NNABP(10))=TEMP                                          WIND 930
      TEMP=DATA(L,J,NNABP(13))*CA+DATA(L,J,NNABP(15))*SA                WIND 940
      DATA(L,J,NNABP(15))=DATA(L,J,NNABP(15))*CA-DATA(L,J,NNABP(13))*SA WIND 950
      DATA(L,J,NNABP(13))=TEMP                                          WIND 960
      TEMP=DATA(L,J,NNABP(16))*CA+DATA(L,J,NNABP(18))*SA                WIND 970
      DATA(L,J,NNABP(18))=DATA(L,J,NNABP(18))*CA-DATA(L,J,NNABP(16))*SA WIND 980
  280 DATA(L,J,NNABP(16))=TEMP                                          WIND 990
  300 CONTINUE                                                          WIND1000
 1001 FORMAT(8F10.4)                                                    WIND1010
 1002 FORMAT(A4,4X,I2)                                                  WIND1020
 2000 FORMAT(1X,A8)                                                     WIND1030
 2001 FORMAT(1X,A8,49HIS NOT A VALID DERIVATIVE NAME FOR THIS TYPE CASE)WIND1040
 2002 FORMAT(5X,10E13.5)                                                WIND1050
 2003 FORMAT(18H ALPHA BREAKPOINTS/5X,10F13.5)                          WIND1060
 2004 FORMAT(17H MACH BREAKPOINTS/5X,10F13.5)                           WIND1070
 2005 FORMAT(18H PARAM BREAKPOINTS/5X,10F13.5)                          WIND1080
      RETURN                                                            WIND1090
      END                                                               WIND1100
                                                                        WIND1110
```

SUBROUTINE TAPERD

Description: Subroutine TAPERD supervises the reading of the input tape and obtains averages of the channels read in. It also writes the output file if desired. It calls TAPEIN, the user-supplied input routine, to do the actual reading of the input tape.

Subroutine listing:

```
      SUBROUTINE TAPERD                                                 TAPE   0
      COMMON /OPTION/ TAPE,DECK,READ                                    TAPE  10
      COMMON /FLCOND/ ALPHA,THETA,Q,V,MACH,IX,IY,IZ,IXZ,W,PHI,CG,KIAS,  TAPE  20
     -     ALT,LONG,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM               TAPE  30
      REAL MACH,IX,IY,IZ,IXZ,KIAS,AVG(40),DATA(40,100)                  TAPE  40
      LOGICAL TAPE,DECK,READ,DELTA(4)                                   TAPE  50
      INTEGER ST(4),ET(4),TIME(4,100),JST(4)                            TAPE  60
      NFRAME=100                                                        TAPE  70
      DO 10 I=1,40                                                      TAPE  80
   10 AVG(I)=0.                                                         TAPE  90
      NPT=0                                                             TAPE 100
   20 CALL TAPEIN(DATA,TIME,NFRAME,ST,ET)                               TAPE 110
      NFR=IABS(NFRAME)                                                  TAPE 120
      DO 100 I=1,NFR                                                    TAPE 130
      IF(TAPE) WRITE (4) (TIME(J,I),J=1,4),(DATA(J,I),J=1,25)           TAPE 140
      DO 30 J=1,40                                                      TAPE 150
   30 AVG(J)=AVG(J)+DATA(J,I)                                           TAPE 160
      NPT=NPT+1                                                         TAPE 170
      IF(NPT.NE.1) GO TO 100                                            TAPE 180
      DO 40 J=1,4                                                       TAPE 190
   40 JST(J)=TIME(J,I)                                                  TAPE 200
  100 CONTINUE                                                          TAPE 210
      IF(NFRAME.GT.0) GO TO 20                                          TAPE 220
  110 ANPT=NPT                                                          TAPE 230
      DO 120 I=1,40                                                     TAPE 240
  120 AVG(I)=AVG(I)/ANPT                                                TAPE 250
      WRITE(3,2000)NPT,JST,(TIME(J,NFR),J=1,4),AVG                      TAPE 260
 2000 FORMAT(1H0,I5,22H POINTS IN CASE, TIMES,4I4,4H  TO,4I4/           TAPE 270
     -     17H0CHANNEL AVERAGES/(10X,10F12.4))                         TAPE 280
      RETURN                                                            TAPE 290
      END                                                               TAPE 300
```

SUBROUTINE PNCH

Description: Subroutine PNCH dimensionalizes coefficients and punches the MMLE card deck.

Programing notes: Through card 540, this subroutine contains some computations and initializations used in all cases. Then cards 590 to 980 contain the lateral-directional dimensionalization and computations; cards 1030 to 1390 contain this information for the longitudinal cases. The remaining cards control the punching of the output deck.

Subroutine listing:

```
      SUBROUTINE PNCH                                                   PNCH   0
      COMMON /FLCOND/ ALPHA,THETA,Q,V,MACH,IX,IY,IZ,IXZ,W,PHI,CG,KIAS,  PNCH  10
     -     ALT,LONG,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM               PNCH  20
      COMMON /COM/ MZLA,MZLO,S,SPAN,CBAR,CGLA,CGLO,METRIC,D1LO,         PNCH  30
     -     D1LA,VEH,APRALA,APRBLA,APRALO,APRBLO,WMLA,WMLO,PUNCH,CORECT, PNCH  40
     -     XALF,XB,ZB,XAN,ZAX,XAY,ZAY,SPS,DLA,DLO                       PNCH  50
      COMMON /DATAWT/ NBP,NMBP,NABP,DATA                                PNCH  60
      REAL MACH,IX,IY,IZ,IXZ,KIAS,AVG(40),DATA(3000),D1LO(7),D1LA(7),   PNCH  70
     -     VEH(2),APRALA(5,4),APRALO(5,4),APRBLA(5,8),APRBLO(5,8),      PNCH  80
     -     A(5,4),B(5,5),BB(5,5),X(21),MASS                             PNCH  90
      INTEGER ST(4),ET(4),FLT,CASE                                      PNCH 100
      LOGICAL LONG,DELTA(4),METRIC,PUNCH,CORECT,DLA,DLO                 PNCH 110
      DATA ALAB,BLAB,BBLAB,ALAT,ALON/1HA,1HB,2HBB,4HLATR,4HLONG/        PNCH 120
      A(5,1)=4.                                                         PNCH 130
      A(5,2)=4.                                                         PNCH 140
      A(5,3)=ALAB                                                       PNCH 150
      B(5,1)=4.                                                         PNCH 160
      B(5,2)=5.                                                         PNCH 170
      B(5,3)=BLAB                                                       PNCH 180
      CALL AZOT(A)                                                      PNCH 190
      CALL AZOT(B)                                                      PNCH 200
      CALL AMAKE(BB,B)                                                  PNCH 210
      BB(5,3)=BBLAB                                                     PNCH 220
      BB(1,5)=1.                                                        PNCH 230
      BB(2,5)=1.                                                        PNCH 240
      BB(4,5)=1.                                                        PNCH 250
      CALL COND                                                         PNCH 260
      TIMESC=.5                                                         PNCH 270
      IT=((ET(1)-ST(1))*3600+(ET(2)-ST(2))*60+ET(3)-ST(3))*1000+        PNCH 280
     -     ET(4)-ST(4)                                                  PNCH 290
      IF(IT.GT.12500) TIMESC=1.                                         PNCH 300
      IF(IT.GT.25000) TIMESC=2.                                         PNCH 310
      CALL INTERP(DATA,NBP,NMBP,NABP,X)                                 PNCH 320
      RAD=57.2958                                                       PNCH 330
      CGFLT=CG                                                          PNCH 340
      IF(CG.NE.999.) GO TO 7                                            PNCH 350
 3007 FORMAT(44HOCG NOT SPECIFIED. DEFAULTED TO WIND TUNNEL.)           PNCH 360
      WRITE(3,3007)                                                     PNCH 370
      CGFLT=CGLA                                                        PNCH 380
      IF(LONG) CGFLT=CGLO                                               PNCH 390
    7 DCG=0.                                                            PNCH 400
      ALPR=ALPHA/RAD                                                    PNCH 410
      THETR=THETA/RAD                                                   PNCH 420
      PHIR=PHI/RAD                                                      PNCH 430
      STH=SIN(THETR)                                                    PNCH 440
      CT=COS(THETR)                                                     PNCH 450
      CP=COS(PHIR)                                                      PNCH 460
      G=32.172                                                          PNCH 470
      IF(METRIC) G=9.80665                                              PNCH 480
      MASS=W/G                                                          PNCH 490
      QS=Q*S                                                            PNCH 500
      QSOMV=QS/(MASS*V)                                                 PNCH 510
      WRITE(2,3000)VEH,FLT,CASE,MACH,ALPHA,PARAM,W,IX,IY,IZ,IXZ,Q,V,    PNCH 520
     -     PUNCH,TIMESC                                                 PNCH 530
      WRITE(3,2000)ALPHA,MACH,Q,V,CGFLT,PARAM                           PNCH 540
      IF(LONG) GO TO 100                                                PNCH 550
C                                                                       PNCH 560
```

```
C      LATERAL                                                          PNCH 570
C                                                                       PNCH 580
       WMAPR=ABS(WMLA)                                                  PNCH 590
       IF(.NOT.CORECT) GO TO 10                                         PNCH 600
       DCG=CGFLT-CGLA                                                   PNCH 610
       XBC=XB                                                           PNCH 620
       XAYC=XAY                                                         PNCH 630
       IF(XB.NE.0.) XBC=XBC+DCG*CBAR                                    PNCH 640
       IF(XAY.NE.0.) XAYC=XAYC+DCG*CBAR                                 PNCH 650
       WRITE(2,3001)XBC,ZB,XAYC,ZAY                                     PNCH 660
   10  TYPE=ALAT                                                        PNCH 670
       BB(3,5)=1.                                                       PNCH 680
       QSOMV=QSOMV*RAD                                                  PNCH 690
       QSB=QS*SPAN*RAD                                                  PNCH 700
       QSBIX=QSB/IX                                                     PNCH 710
       QSBIZ=QSB/IZ                                                     PNCH 720
       B2V=SPAN/(2.*V*RAD)                                              PNCH 730
       QSBVIX=QSBIX*B2V                                                 PNCH 740
       QSBVIZ=QSBIZ*B2V                                                 PNCH 750
       DCG=DCG*CBAR/SPAN                                                PNCH 760
       A(1,1)=QSOMV*X(1)                                                PNCH 770
       A(2,1)=QSBIX*X(2)                                                PNCH 780
       A(3,1)=QSBIZ*(X(3)+DCG*X(1))                                     PNCH 790
       A(1,2)=SIN(ALPR)                                                 PNCH 800
       A(2,2)=QSBVIX*X(4)                                               PNCH 810
       A(3,2)=QSBVIZ*X(5)                                               PNCH 820
       A(4,2)=1.                                                        PNCH 830
       A(1,3)=-COS(ALPR)                                                PNCH 840
       A(2,3)=QSBVIX*X(6)                                               PNCH 850
       A(3,3)=QSBVIZ*X(7)                                               PNCH 860
       A(4,3)=CP*STH/CT                                                 PNCH 870
       A(1,4)=CP*CT*G/V                                                 PNCH 880
       DO 20 I=1,4                                                      PNCH 890
       J=3*I+5                                                          PNCH 900
       B(1,I)=QSOMV*X(J)                                                PNCH 910
       B(2,I)=QSBIX*X(J+1)                                              PNCH 920
       B(3,I)=QSBIZ*(X(J+2)+DCG*X(J))                                   PNCH 930
       IF(.NOT.DELTA(I)) GO TO 20                                       PNCH 940
       BB(1,I)=1.                                                       PNCH 950
       BB(2,I)=1.                                                       PNCH 960
       BB(3,I)=1.                                                       PNCH 970
   20  CONTINUE                                                         PNCH 980
       GO TO 200                                                        PNCH 990
C                                                                       PNCH1000
C      LONGITUDINAL                                                     PNCH1010
C                                                                       PNCH1020
  100  WMAPR=ABS(WMLO)                                                  PNCH1030
       IF(.NOT.CORECT) GO TO 110                                        PNCH1040
       DCG=CGFLT-CGLO                                                   PNCH1050
       XALFC=XALF                                                       PNCH1060
       XANC=XAN                                                         PNCH1070
       IF(XALFC.NE.0.) XALFC=XALFC+DCG*CBAR                             PNCH1080
       IF(XANC.NE.0.) XANC=XANC+DCG*CBAR                                PNCH1090
       WRITE(2,3003)XALFC,XANC,ZAX                                      PNCH1100
  110  TYPE=ALON                                                        PNCH1110
       WRITE(2,3002)                                                    PNCH1120
       QSOM=QSOMV*V                                                     PNCH1130
```

```
      QSCIY=QS*CBAR/IY                                                  PNCH1140
      V2=2./V                                                           PNCH1150
      QSCVIY=QSCIY*CBAR/(2.*V)                                          PNCH1160
      A(1,1)=-QSOMV*X(1)*RAD                                            PNCH1170
      A(2,1)=QSCIY*(X(2)*RAD-DCG*A(1,1)/QSOMV)                          PNCH1180
      A(3,1)=-QSOM*X(3)*RAD                                             PNCH1190
      A(1,2)=1.                                                         PNCH1200
      A(2,2)=QSCVIY*X(4)                                                PNCH1210
      A(4,2)=CP                                                         PNCH1220
      A(1,3)=-QSOMV*V2*X(5)                                             PNCH1230
      A(2,3)=QSCIY*V2*X(6)                                              PNCH1240
      A(3,3)=-QSOM*V2*X(7)                                              PNCH1250
      A(1,4)=-STH*CP*G/V                                                PNCH1260
      A(3,4)=-CT*G                                                      PNCH1270
      DO 130 I=1,4                                                      PNCH1280
      J=3*I+5                                                           PNCH1290
      B(1,I)=-QSOMV*X(J)*RAD                                            PNCH1300
      B(2,I)=QSCIY*(X(J+1)*RAD-DCG*B(1,I)/QSOMV)                        PNCH1310
      B(3,I)=-QSOM*X(J+2)*RAD                                           PNCH1320
      IF(.NOT.DELTA(I)) GO TO 130                                       PNCH1330
      BB(1,I)=1.                                                        PNCH1340
      BB(2,I)=1.                                                        PNCH1350
  130 CONTINUE                                                          PNCH1360
      B(1,5)=-QSOMV*X(20)-(A(1,1)*ALPHA+B(1,1)*DETRIM)/RAD+CP*CT*G/V    PNCH1370
      B(2,5)=-(A(2,1)*ALPHA+B(2,1)*DETRIM)/RAD                          PNCH1380
      B(3,5)=-QSOM*X(21)-(A(3,1)*ALPHA+B(3,1)*DETRIM)/RAD               PNCH1390
  200 IF(WMAPR.EQ.99999.) WMAPR=0.                                      PNCH1400
      WRITE(2,3004)WMAPR,ALPHA,MACH,CGFLT,PARAM,TYPE,S,SPAN,CBAR,SPS,  PNCH1410
     1        ST,ET                                                     PNCH1420
      CALL PMAT(A)                                                      PNCH1430
      CALL PMAT(B)                                                      PNCH1440
      IF(DELTA(1).OR.DELTA(2).OR.DELTA(3).OR.DELTA(4)) CALL PMAT(BB)    PNCH1450
      IF(LONG) GO TO 210                                                PNCH1460
      IF(DLA) WRITE(2,3005)MZLA,DILA                                    PNCH1470
      IF(WMLA.LT.0.) GO TO 250                                          PNCH1480
      CALL PMAT1(APRALA)                                                PNCH1490
      CALL PMAT1(APRBLA)                                                PNCH1500
      GO TO 250                                                         PNCH1510
  210 IF(DLO) WRITE(2,3005)MZLO,DILO                                    PNCH1520
      IF(WMLO.LT.0.) GO TO 250                                          PNCH1530
      CALL PMAT1(APRALO)                                                PNCH1540
      CALL PMAT1(APRBLO)                                                PNCH1550
  250 WRITE(2,3006)                                                     PNCH1560
      RETURN                                                            PNCH1570
 2000 FORMAT(8HOALPHA =,F6.2,9H   MACH =,F5.3,6H   Q =,F7.1,6H   V =,   PNCH1580
     -     F7.1,7H   CG =,F5.3,12H   PARAM =,F10.4)                     PNCH1590
 3000 FORMAT(2A4,6X,6HFLIGHT,I4,6H  CASE,I4,6X,5HMACH=,F5.3,            PNCH1600
     -     8H  ALPHA=,F6.2,8H  PARAM=,F7.2/15H $INPUT GROSWT=,F7.0,     PNCH1610
     -     6H  ,IX=,F7.0,6H  ,IY=,F7.0,6H  ,IZ=,F7.0,7H  ,IXZ=,F7.1,1H,/PNCH1620
     -     3H Q=,F6.1,5H  ,V=,F6.1,9H  ,PUNCH=,L1,10H  ,TIMESC=,F3.1,   PNCH1630
     -     10H  ,BOTH=T,)                                               PNCH1640
 3001 FORMAT(4H XB=,F6.2,6H  ,ZB=,F6.2,7H  ,XAY=,F6.2,7H  ,ZAY=,F6.2,   PNCH1650
     -     1H,)                                                         PNCH1660
 3002 FORMAT(15H ZMAX(3)=1000.,)                                        PNCH1670
 3003 FORMAT(6H XALF=,F6.2,7H  ,XAN=,F6.2,7H  ,ZAX=,F6.2,1H,)           PNCH1680
 3004 FORMAT(7H WMAPR=,E8.2,9H  ,ALPHA=,F6.2,8H  ,MACH=,F5.3,6H  ,CG=,  PNCH1690
     -     F5.3,9H  ,PARAM=,F10.4,1H,/1X,A4,7H=T,  S=,F5.0,             PNCH1700
```

```
     -      8H   ,SPAN=,F6.2,8H   ,CBAR=,F6.2,7H   ,SPS=,F4.0,7H,   $END/     PNCH1710
     -      2(3I2,I3,1X))                                                      PNCH1720
3005 FORMAT(2H01,7X,I1/7F10.1)                                                  PNCH1730
3006 FORMAT(7HENDCASE)                                                          PNCH1740
     END                                                                        PNCH1750
```

SUBROUTINE INTERP

Description: Subroutine INTERP interpolates predicted derivative data tables to obtain the nondimensional derivatives for a particular flight condition.

Programing notes: The subroutine first brackets the Mach number and angle of attack of the flight condition between breakpoints of the predicted data; it also selects the correct set of predicted data depending on the value of PARAM. The interpolation is divided into four sections. The interpolation occurs in one of the four sections on the basis of how many Mach and angle-of-attack breakpoints are specified. If only one breakpoint is specified, the required code changes slightly, because there are not two points to interpolate between.

Subroutine listing:

```
      SUBROUTINE INTERP(DATA,NBP,NMBP,NABP,X)                          INTE   0
C                                                                      INTE  10
C     INTERPOLATES WIND TUNNEL DATA                                    INTE  20
C                                                                      INTE  30
      COMMON /FLCOND/ ALPHA,THETA,Q,V,MACH,IX,IY,IZ,IXZ,W,PHI,CG,KIAS, INTE  40
     -     ALT,LONG,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM              INTE  50
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONGWT                INTE  60
      REAL MACH,IX,IY,IZ,IXZ,KIAS,AVG(40),DATA(NBP,NMBP,1),ABP(16),    INTE  70
     -     MBP(16),BP(8),X(21)                                         INTE  80
      INTEGER ST(4),ET(4),FLT,CASE                                     INTE  90
      LOGICAL DELTA(4),LONG,LONGWT(8),XORF                             INTE 100
C     FIND CORRECT SET OF DATA                                         INTE 110
      L=1                                                              INTE 120
      DO 60 II=1,NBP                                                   INTE 130
      XORF=(LONG.AND.LONGWT(II)).OR.(.NOT.LONG.AND..NOT.LONGWT(II))    INTE 140
   60 IF((PARAM.EQ.BP(II).OR.PARAM*BP(II).EQ.0.) .AND. XORF) L=II      INTE 150
C     BRACKET ALPHA                                                    INTE 160
      IF(NABP.EQ.1) GO TO 50                                           INTE 170
      DO 40 J=2,NABP                                                   INTE 180
      IF(ALPHA.GT. ABP(J)) GO TO 40                                    INTE 190
      EAP=(ALPHA-ABP(J-1))/(ABP(J)-ABP(J-1))                           INTE 200
      IF(EAP.LT.0.) EAP=0.                                             INTE 210
      GO TO 50                                                         INTE 220
   40 CONTINUE                                                         INTE 230
      J=NABP                                                           INTE 240
      EAP=1.                                                           INTE 250
C     BRACKET MACH NUMBER                                              INTE 260
   50 IF(NMBP.EQ.1) GO TO 100                                          INTE 270
      DO 20 I=2,NMBP                                                   INTE 280
      IF(MACH.GT. MBP(I)) GO TO 20                                     INTE 290
      EMN=(MACH-MBP(I-1))/(MBP(I)-MBP(I-1))                            INTE 300
      IF(EMN.LT.0.) EMN=0.                                             INTE 310
      GO TO 30                                                         INTE 320
   20 CONTINUE                                                         INTE 330
      I=NMBP                                                           INTE 340
      EMN=1.                                                           INTE 350
   30 IM1=I-1                                                          INTE 360
      IF(NABP.EQ.1) GO TO 120                                          INTE 370
C     INTERPOLATE DATA                                                 INTE 380
      DO 90 K=1,NCMAX                                                  INTE 390
      JK=(K-1)*NABP+J                                                  INTE 400
      JM1K=JK-1                                                        INTE 410
      PA=(DATA(L,I,JK)-DATA(L,IM1,JK))*EMN+DATA(L,IM1,JK)              INTE 420
      PB=(DATA(L,I,JM1K)-DATA(L,IM1,JM1K))*EMN+DATA(L,IM1,JM1K)        INTE 430
   90 X(K)=(PA-PB)*EAP+PB                                              INTE 440
      GO TO 200                                                        INTE 450
  100 IF(NABP.EQ.1) GO TO 140                                          INTE 460
C     INTERPOLATE IF ONLY 1 MACH BREAKPOINT                            INTE 470
      DO 110 K=1,NCMAX                                                 INTE 480
      JK=(K-1)*NABP+J                                                  INTE 490
      JM1K=JK-1                                                        INTE 500
  110 X(K)=DATA(L,1,JM1K)+EAP*(DATA(L,1,JK)-DATA(L,1,JM1K))            INTE 510
      GO TO 200                                                        INTE 520
C     IF ONLY 1 ALPHA BREAKPOINT                                       INTE 530
  120 DO 130 K=1,NCMAX                                                 INTE 540
  130 X(K)=(DATA(L,I,K)-DATA(L,IM1,K))*EMN+DATA(L,IM1,K)               INTE 550
      GO TO 200                                                        INTE 560
```

```
C     IF ONLY 1 ALPHA AND 1 MACH BREAKPOINT                          INTE 570
  140 DO 150 K=1,NCMAX                                               INTE 580
  150 X(K)=DATA(L,1,K)                                               INTE 590
  200 RETURN                                                         INTE 600
      END                                                            INTE 610
```

SUBROUTINE PMAT

Description: Subroutine PMAT punches a matrix on cards in an 8F10.5 format.

Subroutine listing:

```
      SUBROUTINE PMAT(A)                                             PMAT   0
C     PUNCHES A MATRIX                                               PMAT  10
      COMMON /ALLDIM/ MAX,MIX                                        PMAT  20
      REAL A(1)                                                      PMAT  30
      II=A(MAX)                                                      PMAT  40
      JJ=A(2*MAX)                                                    PMAT  50
      WRITE(2,1000)A(3*MAX),II,JJ                                    PMAT  60
      KE=(JJ-1)*MAX                                                  PMAT  70
      DO 20 I=1,II                                                   PMAT  80
      KEND=I+KE                                                      PMAT  90
   20 WRITE(2,1001)(A(K),K=I,KEND,MAX)                               PMAT 100
      CALL ASPIT(A)                                                  PMAT 110
 1000 FORMAT(A4,4X,I2,I10)                                           PMAT 120
 1001 FORMAT(8F10.5)                                                 PMAT 130
      RETURN                                                         PMAT 140
      END                                                            PMAT 150
```

SUBROUTINE PMAT1

Description: Subroutine PMAT1 punches a matrix on cards in an 8E10.3 format.

Programing notes: This subroutine is needed in addition to PMAT because the APRA and APRB matrices may contain large values but do not need as many significant figures as other matrices.

Subroutine listing:

```
      SUBROUTINE PMAT1(A)                                            PMAT   0
C     PUNCHES A MATRIX IN E FORMAT                                   PMAT  10
      COMMON /ALLDIM/ MAX,MIX                                        PMAT  20
      REAL A(1)                                                      PMAT  30
      II=A(MAX)                                                      PMAT  40
      JJ=A(2*MAX)                                                    PMAT  50
      WRITE(2,1000)A(3*MAX),II,JJ                                    PMAT  60
      KE=(JJ-1)*MAX                                                  PMAT  70
      DO 20 I=1,II                                                   PMAT  80
      KEND=I+KE                                                      PMAT  90
   20 WRITE(2,1001)(A(K),K=I,KEND,MAX)                               PMAT 100
 1000 FORMAT(A4,4X,I2,I10)                                           PMAT 110
 1001 FORMAT(8E10.3)                                                 PMAT 120
      RETURN                                                         PMAT 130
      END                                                            PMAT 140
```

SUBROUTINE RDSET

Description: Subroutine RDSET is user supplied; the subroutine listed here is a sample. This subroutine should do any initialization or input required before calling subroutine TAPEIN.

Subroutine listing:

```
      SUBROUTINE RDSET                                                  RDSE   1
C                                                                       RDSE  10
C     THIS SUBROUTINE SHOULD INCLUDE ANY INITIALIZATION DESIRED FOR     RDSE  20
C     READING THE INPUT TAPE, FOR INSTANCE SPECIFYING CHANNEL NUMBERS.  RDSE  30
C     DATA SHOULD BE PASSED TO SUBROUTINE TAPEIN WITH LABELLED COMMON   RDSE  40
C          BLOCK /TAPDAT/                                               RDSE  50
C     THIS SAMPLE VERSION READS THE NUMBER OF WORDS ON THE INPUT TAPE   RDSE  60
C     AND THE CHANNEL NUMBERS OFTHE SIGNALS NEEDED                      RDSE  70
C                                                                       RDSE  80
      COMMON /TAPDAT/ NWORD,ICHAN                                       RDSE  90
      INTEGER ICHAN(40)                                                 RDSE 100
      READ (1,1000) NWORD                                               RDSE 110
      READ (1,1000) ICHAN                                               RDSE 120
      WRITE(3,2000)NWORD,ICHAN                                          RDSE 130
 1000 FORMAT(16I5)                                                      RDSE 140
 2000 FORMAT(20H0INPUT FILE CONTAINS,I5,22H DATA WORDS PER RECORD/      RDSE 150
          9H CHANNELS/(10X,20I5))                                       RDSE 160
      RETURN                                                            RDSE 170
      END                                                               RDSE 180
```

SUBROUTINE TAPEIN

Description: Subroutine TAPEIN is user supplied; the subroutine listed here is a sample. This subroutine should be written to read data in the form available for a particular flight program. The comment cards and sample program illustrate the conventions required for interface with the rest of the program.

Subroutine listing:

```
      SUBROUTINE TAPEIN(DATA,TIME,NFRAME,ST,ET)                          TAPE   0
C                                                                        TAPE  10
C     THIS SUBROUTINE SHOULD READ THE INPUT TAPE AND PLACE UP TO         TAPE  20
C     NFRAME FRAMES IN THE TIME INTERVAL BETWEEN ST AND ET (START TIME   TAPE  30
C     AND END TIME) INTO THE ARRAYS TIME AND DATA                        TAPE  40
C     THE TIME ARRAY SHOULD CONTAIN HOURS,MINUTES,SECONDS,MILLISECONDS   TAPE  50
C     THE DATA ARRAY SHOULD CONTAIN THE DATA CHANNELS IN THE ORDER TO     TAPE  60
C     WRITTEN ON THE OUTPUT TAPE                                         TAPE  70
C     WHEN THE LAST TIME IN THE REQUESTED INTERVAL IS FOUND,             TAPE  80
C     NFRAME SHOULD BE SET TO MINUS THE NUMBER OF FRAMES OF DATA         TAPE  90
C     BEING RETURNED                                                     TAPE 100
C                                                                        TAPE 110
C     THIS SAMPLE VERSION READS AN UNFORMATTED TAPE AND PICKS THE        TAPE 120
C     SIGNALS DESIRED FROM THE CHANNELS SPECIFIED IN SUBROUTINE RDSET    TAPE 130
C                                                                        TAPE 140
      COMMON /TAPDAT/ NWORD,ICHAN                                        TAPE 150
      INTEGER ST(4),ET(4),TIME(4,100),ICHAN(40),IT(4)                    TAPE 160
      REAL DATA(40,100),RECORD(150)                                      TAPE 170
      IST=ST(4)+1000*(ST(3)+60*ST(2)+3600*ST(1))                         TAPE 180
      IET=ET(4)+1000*(ET(3)+60*ET(2)+3600*ET(1))                         TAPE 190
      I=0                                                                TAPE 200
   10 READ (15)IT,(RECORD(J),J=1,NWORD)                                  TAPE 210
      ITM=IT(4)+1000*(IT(3)+60*IT(2)+3600*IT(1))                         TAPE 220
      IF(ITM.LT.IST) GO TO 10                                            TAPE 230
   20 I=I+1                                                              TAPE 240
      DO 30 J=1,4                                                        TAPE 250
   30 TIME(J,I)=IT(J)                                                    TAPE 260
      DO 40 J=1,40                                                       TAPE 270
      DATA(J,I)=0.                                                       TAPE 280
      IF(ICHAN(J).EQ.0) GO TO 40                                         TAPE 290
      DATA(J,I)=RECORD(ICHAN(J))                                         TAPE 300
   40 CONTINUE                                                           TAPE 310
      IF(I.GE.NFRAME) RETURN                                             TAPE 320
      IF(ITM.GE.IET) GO TO 100                                           TAPE 330
      READ (15)IT,(RECORD(J),J=1,NWORD)                                  TAPE 340
      ITM=IT(4)+1000*(IT(3)+60*IT(2)+3600*IT(1))                         TAPE 350
      GO TO 20                                                           TAPE 360
  100 NFRAME=-I                                                          TAPE 370
      RETURN                                                             TAPE 380
      END                                                                TAPE 390
```

SUBROUTINE COND1

Description: Subroutine COND1 is user supplied, and is described by the comment cards.

Subroutine listing:

```
      SUBROUTINE COND1                                                  COND   0
C                                                                       COND  10
C     THIS SUBROUTINE SHOULD INCLUDE ANY INITIALIZATION NEEDED          COND  20
C     FOR SUBROUTINE COND TO DETERMINE THE FLIGHT CONDITION             COND  30
C     TYPICAL ITEMS INCLUDED HERE MIGHT BE TABLES OF INERTIAS AS A      COND  40
C     FUNCTION OF GROSS WEIGHT                                          COND  50
C     ANY DATA MAY BE PASSED TO SUBROUTINE COND THROUGH A LABELLED      COND  60
C     COMMON BLOCK /TABLE/                                              COND  70
C     SUBROUTINE SUPPLIED IS A NULL SUBROUTINE                          COND  80
C                                                                       COND  90
      RETURN                                                            COND 100
      END                                                               COND 110
```

SUBROUTINE COND

Description: Subroutine COND is user supplied. It automatically obtains the flight condition from the channel averages computed by TAPERD. The subroutine listed illustrates the method of doing this.

Subroutine listing:

```
      SUBROUTINE COND                                                   COND   0
C                                                                       COND  10
C     THIS SUBROUTINE SHOULD SPECIFY THE FLIGHT CONDITION AND OTHER     COND  20
C     PARAMETERS NOT READ IN THROUGH NAMELIST /COND/                    COND  30
C     AVG CONTAINS THE AVERAGE VALUES OF EACH CHANNEL READ OFF THE INPUTCOND  40
C     TAPE IF THERE WAS ONE READ                                        COND  50
C     THE USER MAY CHOOSE TO USE THESE AVERAGE VALUES FOR THE FLIGHT    COND  60
C     CONDITION INSTEAD OF READING IT IN                                COND  70
C     FOR INSTANCE, IF ALPHA IS TO BE OBTAINED FROM THE CHANNEL AVERAGE COND  80
C     THE STATEMENT                                                     COND  90
C     ALPHA=AVG(1)                                                      COND 100
C     WOULD BE INCLUDED HERE                                            COND 110
C     THE SEVERAL EXTRA CHANNELS AVAILABLE MAY BE USED TO OBTAIN        COND 120
C     FUEL WEIGHTS OR OTHER QUANTITIES NEEDED TO COMPUTE THE INERTIAS   COND 130
C                                                                       COND 140
C     THE SUBROUTINE SUPPLIED OBTAINS ALPHA,THETA,PHI,DETRIM,Q,V,AND    COND 150
C     MACH FROM SIGNAL AVERAGES AND COMPUTES Q AND V FROM ALTITUDE      COND 160
C     AND KIAS(KNOTS INDICATED AIRSPEED) IF THESE ARE MORE READILY      COND 170
C     AVAILABLE (INDICATED BY A NON-ZERO VALUE OF KIAS)                 COND 180
C                                                                       COND 190
      COMMON /OPTION/ TAPE,DECK,READ                                    COND 200
      COMMON /FLCOND/ ALPHA,THETA,Q,V,MACH,IX,IY,IZ,IXZ,W,PHI,CG,KIAS,  COND 210
     -     ALT,LONG,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM               COND 220
      REAL MACH,IX,IY,IZ,IXZ,KIAS,AVG(40)                               COND 230
      INTEGER ST(4),ET(4),FLT,CASE                                      COND 240
      LOGICAL DELTA(4),LONG,TAPE,DECK,READ                              COND 250
      IF(.NOT.READ) GO TO 10                                            COND 260
      ALPHA=AVG(1)                                                      COND 270
      THETA=AVG(4)                                                      COND 280
      PHI=AVG(12)                                                       COND 290
      DETRIM=AVG(8)                                                     COND 300
      Q=AVG(15)                                                         COND 310
      V=AVG(3)                                                          COND 320
      MACH=AVG(14)                                                      COND 330
   10 IF(KIAS.EQ.0.) RETURN                                             COND 340
      Q=(KIAS*.0582)**2                                                 COND 350
      DALT=ALT*.001                                                     COND 360
      V=1.688*KIAS*EXP(DALT*(.01375+.000C975*DALT))                     COND 370
      RETURN                                                            COND 380
      END                                                               COND 390
```

SUBROUTINE LOAD1

Description: Subroutine LOAD1 reads a matrix from cards.

Subroutine listing:

```
      SUBROUTINE LOAD1(A)                                          LOAD   0
      COMMON /ALLDIM/ MAX,MIX                                      LOAD  10
      REAL A(1)                                                    LOAD  20
      MAX3=3*MAX                                                   LOAD  30
      READ (1,1000) A(MAX3),II,JJ                                  LOAD  40
      A(MAX)=II                                                    LOAD  50
      A(2*MAX)=JJ                                                  LOAD  60
      KE=(JJ-1)*MAX                                                LOAD  70
      DO 10 I=1,II                                                 LOAD  80
      KEND=I+KE                                                    LOAD  90
   10 READ (1,1001) (A(K),K=I,KEND,MAX)                            LOAD 100
 1000 FORMAT(A4,4X,I2,I10)                                         LOAD 110
 1001 FORMAT(8F10.4)                                               LOAD 120
      RETURN                                                       LOAD 130
      END                                                          LOAD 140
```

SUBROUTINES ASPIT, AMAKE, AND AZOT

Subroutines ASPIT, AMAKE, and AZOT are identical to those used in the MMLE program.

APPENDIX D

SAMPLE CASE FOR THE SETUP PROGRAM

This appendix presents a sample check case for the SETUP program.

INPUT CARDS

```
PUNCH DECK
START
 $WIND NMBP=2,NARP=2,NCLO=8,LONG(1)=T,SPAN=15.,CBAR=6.,S=100.,$END
SAMPLE
 0.
CL        2
 .1         .5
0.          .4
CD        2
 .05        .1
  .07       .12
CLA       1
 .07        .065
CDA       1
 .01        .015
CMA       1
 -.005      -.006
CLDE      1
 .01        .01
CMDE      1
 -.009      -.011
CMQ       1
 -5.        -5.
 .3         .7
 .0         5.
0.
102051000 102100000 SAMPLE CASE 1
 $COND IX=300.,IY=2000.,IZ=2010.,IXZ=10.,W=2500.,LONG=T,
 FLT=1,CASE=1,Q=50.,V=450.,ALPHA=4.,MACH=.5,$END
102512000 102522000 SAMPLE CASE 2
 $COND CASE=2,Q=60.,V=500.,ALPHA=3.,MACH=.55,$END
-1
```

OUTPUT LISTING

```
SAMPLE        WIND TUNNEL DATA.  REF CG = .250 (LAT), .250 (LONG)    PER RADIAN? F
CL
       .10000E+00    .50000E+00
      0.             .40000E+00
CD
       .50000E-01    .10000E+00
       .70000E-01    .12000E+00
CLA
       .70000E-01    .65000E-01
CDA
       .10000E-01    .15000E-01
CMA
      -.50000E-02   -.60000E-02
CLDE
       .10000E-01    .10000E-01
CMDE
      -.90000E-02   -.11000E-01
CMQ
      -.50000E+01   -.50000E+01
ALPHA BREAKPOINTS
           .30000        .70000
MACH BREAKPOINTS
          0.00000       5.00000
PARAM BREAKPOINTS
          0.00000
```

```
                    FLIGHT  1      CASE    1      TIME  10  20  51   0  TO  10  21   0   0      LONGITUDINAL? T

CG NOT SPECIFIED. DEFAULTED TO WIND TUNNEL.

ALPHA =   4.00    MACH = .500    Q =    50.0    V =   450.0    CG = .250    PARAM =     0.0000
A                                              4  BY  4
              -.5841E+00    .1000E+01 -0.          -0.
              -.4383E+01   -.5000E+00  0.           0.
              -.3970E+01   0.          -0.          -.3217E+02
              0.            .1000E+01  0.           0.
B                                              4  BY  5
              -.8192E-01 -0.           -0.          -0.            .4204E-01
              -.7907E+01  0.           0.           0.            .3060E+00
               .4504E+00 -0.           -0.          -0.           -.5900E+01
              0.          0.           0.           0.           0.
```

```
                    FLIGHT  1      CASE   2      TIME  10  25  12   0  TO  10  25  22   0      LONGITUDINAL? T

CG NOT SPECIFIED. DEFAULTED TO WIND TUNNEL.

ALPHA =   3.00    MACH = .550    Q =    60.0    V =   500.0    CG = .250    PARAM =     0.0000
A                                      4  BY  4
              -.6304E+00    .1000E+01 -0.          -0.
              -.5270E+01   -.5400E+00  0.           0.
              -.5065E+01    0.         -0.           -.3217E+02
              0.             .1000E+01  0.           0.
B                                      4  BY  5
              -.8847E-01 -0.           -0.          -0.            .2165E-01
              -.9509E+01   0.           0.           0.            .2759E+00
               .5405E+00  -0.          -0.          -0.           -.7164E+01
              0.           0.           0.           0.           0.
```

PUNCHED CARD OUTPUT LISTING

```
SAMPLE        FLIGHT   1  CASE   1       MACH= .500  ALPHA=  4.00  PARAM=   0.00
 $INPUT GROSWT=  2500.  ,IX=   300.  ,IY=  2000.  ,IZ=  2000.  ,IXZ=   10.0,
 Q=  50.0  ,V= 450.0  ,PUNCH=F  ,TIMESC= .5  ,BOTH=T,
 ZMAX(3)=1000.,
 WMAPR=0.        ,ALPHA=  4.00  ,MACH= .500  ,CG= .250  ,PARAM=    0.0000,
 LONG=T,  S= 100.  ,SPAN= 15.00  ,CBAR=  6.00  ,SPS=  0.,  $END
102051  0 1021 0  0
A         4          4
   -.58412   1.00000  -0.00000  -0.00000
  -4.38313   -.50000   0.00000   0.00000
  -3.97020   0.00000  -0.00000 -32.17200
   0.00000   1.00000   0.00000   0.00000
B         4          5
   -.08192  -0.00000  -0.00000  -0.00000    .04204
  -7.90682   0.00000   0.00000   0.00000    .30600
    .45040  -0.00000  -0.00000  -0.00000  -5.90024
   0.00000   0.00000   0.00000   0.00000   0.00000
ENDCASE
SAMPLE        FLIGHT   1  CASE   2       MACH= .550  ALPHA=  3.00  PARAM=   0.00
 $INPUT GROSWT=  2500.  ,IX=   300.  ,IY=  2000.  ,IZ=  2000.  ,IXZ=   10.0,
 Q=  60.0  ,V= 500.0  ,PUNCH=F  ,TIMESC= .5  ,BOTH=T,
 ZMAX(3)=1000.,
 WMAPR=0.        ,ALPHA=  3.00  ,MACH= .550  ,CG= .250  ,PARAM=    0.0000,
 LONG=T,  S= 100.  ,SPAN= 15.00  ,CBAR=  6.00  ,SPS=  0.,  $END
102512  0 102522  0
A         4          4
   -.63044   1.00000  -0.00000  -0.00000
  -5.27007   -.54000   0.00000   0.00000
  -5.06514   0.00000  -0.00000 -32.17200
   0.00000   1.00000   0.00000   0.00000
B         4          5
   -.08847  -0.00000  -0.00000  -0.00000    .02165
  -9.50881   0.00000   0.00000   0.00000    .27594
    .54048  -0.00000  -0.00000  -0.00000  -7.16407
   0.00000   0.00000   0.00000   0.00000   0.00000
ENDCASE
```

APPENDIX E

SUMARY PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the SUMARY program are presented together with supplemental information.

MAIN PROGRAM SUMARY

Description: The main program SUMARY sets defaults, reads the NAMELIST, and initializes variables.

Program listing:

```
      PROGRAM SUMARY(INPUT,OUTPUT,TAPE69,TAPE1=INPUT,TAPE3=OUTPUT)      MAIN   0
C                                                                        MAIN  10
C     SUMMARY PLOT PROGRAM FOR MMLE DATA                                 MAIN  20
C                                                                        MAIN  30
      COMMON /ALLDIM/ MAX,MIX                                            MAIN  40
      COMMON /LINCOM/ HGT                                                MAIN  50
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                    MAIN  60
      COMMON /CGCOR/ SHIFT,CGLA,CGLO,COB                                 MAIN  70
      COMMON /NBPS/ NMBP,NABP,NBP,NPARAM                                 MAIN  80
      COMMON /INS/ NPLOT,WTPLOT                                          MAIN  90
      COMMON /SUMDAT/ YLOC,XSKIP,ALEN,ASCAL2,YSTEP,AMIN,TABLAB,FDATA,    MAIN 100
     -     FDATAC,TITLE                                                  MAIN 110
      COMMON /PSCL/ CREACT,IWT1,IWT2,YLEN2,II,NPARM                      MAIN 120
      REAL TITLE(20),ABP(16),BP(8),MBP(16),DATA(3000),FDATA(5000),       MAIN 130
     -     FDATAC(5000),BUF(1024),FDAT(200),FDATC(200),ALFS(200),        MAIN 140
     -     TABLAB(2),MLAB(2),PLAB(2)                                     MAIN 150
      LOGICAL PRINT,LONG(8),LATR(8),DEG,RAD,BODY,STAB,SHIFT,WTPLOT       MAIN 160
      DATA MLAB/4HMACH,1H /,PLAB/4HPARA,1HM/                             MAIN 170
      NAMELIST /WIND/ NBP,NABP,NMBP,NCLA,NCLO,RAD,DEG,BODY,STAB,         MAIN 180
     -     LONG,LATR,PRINT,CGLA,CGLO,NPARAM,SHIFT,WTPLOT,CREACT,         MAIN 190
     -     AMIN,AMAX,ASCALE,YLEN,XOIST,CBAR,SPAN                         MAIN 200
      NBUF=1024                                                          MAIN 210
      MAX=4                                                              MAIN 220
      READ (1,1000) TITLE                                                MAIN 230
      WRITE(3,2000)TITLE                                                 MAIN 240
      HGT=.07                                                            MAIN 250
      SHIFT=.FALSE.                                                      MAIN 260
      CBAR=0.                                                            MAIN 270
      SPAN=1.E+50                                                        MAIN 280
      NPARAM=0                                                           MAIN 290
      NCLA=0                                                             MAIN 300
      NCLO=0                                                             MAIN 310
      NBP=1                                                              MAIN 320
      CGLA=.25                                                           MAIN 330
      CGLO=.25                                                           MAIN 340
      NABP=1                                                             MAIN 350
      NMBP=1                                                             MAIN 360
      PRINT=.FALSE.                                                      MAIN 370
      DO 5 I=1,3000                                                      MAIN 380
    5 DATA(I)=0.                                                         MAIN 390
      DO 10 I=1,8                                                        MAIN 400
      LATR(I)=.FALSE.                                                    MAIN 410
      LONG(I)=.TRUE.                                                     MAIN 420
   10 BP(I)=0.                                                           MAIN 430
      RAD=.FALSE.                                                        MAIN 440
      STAB=.TRUE.                                                        MAIN 450
      BODY=.FALSE.                                                       MAIN 460
      WTPLOT=.TRUE.                                                      MAIN 470
      CREACT=1.                                                          MAIN 480
      AMIN=0.                                                            MAIN 490
      AMAX=12.                                                           MAIN 500
      ASCALE=1.                                                          MAIN 510
      YLEN=10.                                                           MAIN 520
      XOIST=10.                                                          MAIN 530
      READ (1,WIND)                                                      MAIN 540
      NPARM=NPARAM                                                       MAIN 550
      COB=CBAR/SPAN                                                      MAIN 560
```

```
      YLEN2=YLEN/2.                                                    MAIN 570
      YLOC=0.                                                          MAIN 580
      ASCAL2=ASCALE*2.                                                 MAIN 590
      ALEN=(AMAX-AMIN)/ASCAL2                                          MAIN 600
      XSKIP=ALEN+XDIST/2.                                              MAIN 610
      YSTEP=YLEN2+1.                                                   MAIN 620
      DO 20 I=1,8                                                      MAIN 630
   20 IF(LATR(I)) LONG(I)=.FALSE.                                      MAIN 640
C     READ WIND TUNNEL DATA                                            MAIN 650
      CALL WINDIN(DATA,NBP,NMBP,NABP,BODY,PRINT,RAD)                   MAIN 660
      IF(SHIFT) WRITE(3,2001)CGLA,CGLO                                 MAIN 670
      IF(CRFACT.NE.0.) WRITE(3,2002)CRFACT                             MAIN 680
      IWT1=1                                                           MAIN 690
      IWT2=1                                                           MAIN 700
      DO 15 I=1,NABP                                                   MAIN 710
      IF(ABP(I).LT.AMIN) IWT1=I+1                                      MAIN 720
   15 IF(ABP(I).LE.AMAX) IWT2=I                                        MAIN 730
      NCMX=NCMAX+2                                                     MAIN 740
      ND=NMBP*2                                                        MAIN 750
      TABLAB(1)=MLAB(1)                                                MAIN 760
      TABLAB(2)=MLAB(2)                                                MAIN 770
      IF(NPARAM.LE.0) GO TO 25                                         MAIN 780
      ND=NPARAM*2                                                      MAIN 790
      TABLAB(1)=PLAB(1)                                                MAIN 800
      TABLAB(2)=PLAB(2)                                                MAIN 810
   25 ND2=ND/2                                                         MAIN 820
C     READ FLIGHT DATA                                                 MAIN 830
      CALL FLIGHT(NCMX,ND,FDATA,FDATAC)                                MAIN 840
      CALL PLOTS(BUF,NBUF,69)                                          MAIN 850
      CALL FACTOR(.787402)                                             MAIN 860
      CALL PLOT(0.,.5,-3)                                              MAIN 870
C     READ PLOTTING INSTRUCTIONS                                       MAIN 880
   30 CALL INSTR                                                       MAIN 890
      IF(NPLOT.LE.0) GO TO 50                                          MAIN 900
C     MAKE PLOTS                                                       MAIN 910
      DO 40 II=1,NPLOT                                                 MAIN 920
   40 CALL SUMPLT(FDAT,FDATC,ALFS,ND2,DATA,NBP,NMBP,NABP)              MAIN 930
      GO TO 30                                                         MAIN 940
   50 CALL PLOT(0.,0.,999)                                             MAIN 950
 1000 FORMAT(20A4)                                                     MAIN 960
 2000 FORMAT(51H1MMLE SUMMARY PLOTTING PROGRAM    ****   1 MAY 1974/  MAIN 970
     -      1H0,20X,20A4)                                             MAIN 980
 2001 FORMAT(55HOCNB AND CMA WILL BE CORRECTED TO THE WIND TUNNEL REFER,MAIN 990
     -      8HENCE CGS,F10.3,7H (LATR),F10.3,7H (LONG))               MAIN1000
 2002 FORMAT(48HOCONFIDENCE LEVELS WILL BE PLOTTED MULTIPLIED BY,F5.1) MAIN1010
      STOP                                                             MAIN1020
      END                                                              MAIN1030
```

SUBROUTINE FLIGHT

Description: Subroutine FLIGHT reads and sorts flight data.

Programing notes: Data are stored in the arrays FDATA and FDATAC. The FDATA array contains derivative values, and the FDATAC array contains confidence levels. Note that the sign of the X and Z coefficients is changed for longitudinal data to agree with the more common N and A (axial) coefficients. The flight C_{m_α} and C_{n_β} are shifted to the wind-tunnel reference center of gravity if SHIFT = T. LONLOC and LATLOC give the positions of data in the A and B matrices considered as vectors.

Subroutine listing:

```
      SUBROUTINE FLIGHT(NCMX,ND,FDATA,FDATAC)                           FLIG   0
C     READS FLIGHT DATA AND SORTS BY MACH OR PARAM                      FLIG  10
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                   FLIG  20
      COMMON /CGCOR/ SHIFT,CGLA,CGLO,COB                                FLIG  30
      COMMON /CASES/ NCASE                                              FLIG  40
      COMMON /NBPS/ NMBP,NABP,NBP,NPARAM                                FLIG  50
      REAL FDATA(NCMX,ND,1),FDATAC(NCMX,ND,1),ABP(16),MBP(16),BP(9),    FLIG  60
     -     A(16),B(32),AC(16),BC(32),MACH,TITL(9)                       FLIG  70
      LOGICAL LONG(8),SHIFT                                             FLIG  80
      INTEGER NCASE(32),LONLOC(21),LATLOC(19)                           FLIG  90
      DATA PLT/4HPLOT/,ALAT/4HLATR/,LONLOC/1,2,3,6,9,10,11,1,2,3,5,6,7, FLIG 100
     -     9,10,11,13,14,15,17,18/,LATLOC/1,2,3,6,7,10,11,1,2,3,5,6,7,  FLIG 110
     -     9,10,11,13,14,15/                                            FLIG 120
      ND2=ND/2                                                          FLIG 130
      DO 10 I=1,32                                                      FLIG 140
   10 NCASE(I)=0                                                        FLIG 150
   20 READ (1,1000) TYPE,TITL,MACH,ALPHA,PARAM,CG                       FLIG 160
      IF(TYPE.EQ.PLT) RETURN                                            FLIG 170
      WRITE(3,2000)TYPE,TITL,MACH,ALPHA,PARAM,CG                        FLIG 180
C     FLIGHT DATA IS STORED AS FDATA(COEFFICIENT,GROUP,CASE)            FLIG 190
C          WHERE GROUP=INDEX IF LONG, OR INDEX+ND/2 IF LATR             FLIG 200
C          AND INDEX IDENTIFIES EITHER THE MACH OR (IF NPARAM.GT.0)     FLIG 210
C          THE EXTRA PARAMETER                                          FLIG 220
      CALL LOAD1(A )                                                    FLIG 230
      CALL LOAD1(B )                                                    FLIG 240
      CALL LOAD1(AC)                                                    FLIG 250
      CALL LOAD1(BC)                                                    FLIG 260
      INDEX=2                                                           FLIG 270
      IF(ND2.LT.2) GO TO 60                                             FLIG 280
      IF(NPARAM.GT.0) GO TO 40                                          FLIG 290
      DO 30 INDEX=2,NMBP                                                FLIG 300
      IF(MACH.LT.(MBP(INDEX)+MBP(INDEX-1))*.5) GO TO 60                 FLIG 310
   30 CONTINUE                                                          FLIG 320
      INDEX=NMBP+1                                                      FLIG 330
      GO TO 60                                                          FLIG 340
   40 DO 50 INDEX=2,NPARAM                                              FLIG 350
      IF(PARAM.LT.(BP(INDEX)+BP(INDEX-1))*.5) GO TO 60                  FLIG 360
   50 CONTINUE                                                          FLIG 370
      INDEX=NPARAM+1                                                    FLIG 380
   60 INDEX=INDEX-1                                                     FLIG 390
      IF(TYPE.EQ.ALAT) GO TO 110                                        FLIG 400
C     LONGITUDINAL - CHANGE SIGN OF X AND Z DERIVATIVES                 FLIG 410
      DO 70 I=1,11,2                                                    FLIG 420
   70 A(I)=-A(I)                                                        FLIG 430
      DO 80 I=1,19,2                                                    FLIG 440
   80 B(I)=-B(I)                                                        FLIG 450
      NCAS=NCASE(INDEX)+1                                               FLIG 460
      IF(SHIFT) A(2)=A(2)+(CGLO-CG)*A(1)                                FLIG 470
      DO 100 I=1,21                                                     FLIG 480
      IF(I.GE.8) GO TO 90                                               FLIG 490
      FDATA(I,INDEX,NCAS)=A(LONLOC(I))                                  FLIG 500
      FDATAC(I,INDEX,NCAS)=AC(LONLOC(I))                                FLIG 510
      GO TO 100                                                         FLIG 520
   90 FDATA(I,INDEX,NCAS)=B(LONLOC(I))                                  FLIG 530
      FDATAC(I,INDEX,NCAS)=BC(LONLOC(I))                                FLIG 540
  100 CONTINUE                                                          FLIG 550
      GO TO 150                                                         FLIG 560
```

```
  110 INDEX=INDEX+ND2                                                    FLIG 570
      NCAS=NCASE(INDEX)+1                                                FLIG 580
      IF(SHIFT) A(3)=A(3)+(CGLA-CG)*COB*A(1)                             FLIG 590
      DO 130 I=1,19                                                      FLIG 610
      IF(I.GE.8) GO TO 120                                               FLIG 610
      FDATA(I,INDEX,NCAS)=A(LATLOC(I))                                   FLIG 620
      FDATAC(I,INDEX,NCAS)=AC(LATLOC(I))                                 FLIG 630
      GO TO 130                                                          FLIG 640
  120 FDATA(I,INDEX,NCAS)=B(LATLOC(I))                                   FLIG 650
      FDATAC(I,INDEX,NCAS)=BC(LATLOC(I))                                 FLIG 660
  130 CONTINUE                                                           FLIG 670
  150 NCASE(INDEX)=NCAS                                                  FLIG 680
      FDATA(22,INDEX,NCAS)=ALPHA                                         FLIG 690
      FDATA(23,INDEX,NCAS)=MACH                                          FLIG 700
      FDATAC(22,INDEX,NCAS)=PARAM                                        FLIG 710
      FDATAC(23,INDEX,NCAS)=CG                                           FLIG 720
      GO TO 20                                                           FLIG 730
 1000 FORMAT(10A4,4F10.4)                                                FLIG 740
 2000 FORMAT(1HO,A4,5X,9A4,4F10.4)                                       FLIG 750
      END                                                                FLIG 760
```

SUBROUTINE INSTR

Description: Subroutine INSTR reads plotting instructions.

Programing notes: The instructions are passed to the rest of the program in the following form:

NPLOT – number of coefficients to be plotted.

LATLON – 1 if lateral data, 2 if longitudinal data.

PARM, TOL – parameter value and tolerance.

LL – number of the predicted derivative data set corresponding to LATLON and PARM.

IDER – parameter numbers that correspond to the coefficients to be plotted.

YMIN, YMAX – minimum and maximum values for the ordinates.

Subroutine listing:

```
      SUBROUTINE INSTR                                                  INST   0
C     READS INSTRUCTIONS ON COEFFICIENTS TO PLOT,SCALES TO USE,         INST  10
C          AND THE PARAMETER AND TOLERANCE FOR FLIGHT POINTS            INST  20
C     DO NOT OVERLAY THIS SUBROUTINE AS START,DERIV,SMIN AND SMAX MUST  INST  30
C          BE PRESERVED                                                 INST  40
      COMMON /NBPS/ NMBP,NABP,NBP,NPARAM                                INST  50
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                   INST  60
      COMMON /INS/ NPLOT,WTPLOT                                         INST  70
      COMMON /SELECT/ PARAM,TOL,IDER,YMIN,YMAX,LATLON,DERIVS,LL,WTPL    INST  80
      REAL DERIV(4),SMIN(4),SMAX(4),CER(21,2),YMIN(21),YMAX(21),        INST  90
     -     DERIVS(21),ABP(16),MBP(16),BP(8)                           INST 100
      INTEGER IDER(21)                                                  INST 110
      LOGICAL WTPLOT,WTPL,LONG(8)                                       INST 120
      DATA END,ALON,ALAT,BLANK,STAR/3HEND,4HLONG,4HLATR,1H ,4HSTAR/     INST 130
      DATA DER/3HCYB,3HCLB,3HCNB,3HCLP,3HCNP,3HCLR,3HCNR,4HCYDA,4HCLDA, INST 140
     -     4HCNDA,4HCYDR,4HCLDR,4HCNCR,4HCYD1,4HCLD1,4HCND1,4HCYD2,     INST 150
     -     4HCLD2,4HCND2,2*1H ,                                         INST 160
     -     3HCNA,3HCMA,3HCAA,3HCMQ,3HCNV,3HCMV,3HCAV,4HCNDE,4HCMDE,     INST 170
     -     4HCADE,4HCNDC,4HCMDC,4HCADC,4HCND1,4HCMD1,4HCAD1,4HCND2,     INST 180
     -     4HCMD2,4HCAD2,2HCN,2HDE/                                     INST 190
      IF(START.NE.STAR) READ (1,1001) DERIV(1),SMIN(1),SMAX(1)          INST 200
      START=STAR                                                        INST 210
      NPLOT=0                                                           INST 220
      IF(DERIV(1).EQ.END) GO TO 120                                     INST 230
      LATLON=1                                                          INST 240
      IF(DERIV(1).EQ.ALON) LATLON=2                                     INST 250
      NC=19                                                             INST 260
      IF(LATLON.EQ.2) NC=21                                             INST 270
      PARAM=SMIN(1)                                                     INST 280
      TOL=SMAX(1)                                                       INST 290
      WRITE(3,2004)DERIV(1),PARAM,TOL                                   INST 300
   20 READ (1,1001) (DERIV(I),SMIN(I),SMAX(I),I=1,4)                    INST 310
      IF(DERIV(1).EQ.ALAT.OR.DERIV(1).EQ.ALON.OR.DERIV(1).EQ.END)GOTO 90INST 320
      DO 70 I=1,4                                                       INST 330
      IF(DERIV(I).EQ.BLANK) GO TO 80                                    INST 340
      NPLOT=NPLOT+1                                                     INST 350
      YMIN(NPLOT)=SMIN(I)                                               INST 360
      YMAX(NPLOT)=SMAX(I)                                               INST 370
      DO 30 J=1,NC                                                      INST 380
      IF(DERIV(I).EQ.DER(J,LATLON)) GO TO 60                            INST 390
   30 CONTINUE                                                          INST 400
      WRITE(3,2002)DERIV(I)                                             INST 410
      STOP                                                              INST 420
   60 IDER(NPLOT)=J                                                     INST 430
   70 DERIVS(NPLOT)=DERIV(I)                                            INST 440
      GO TO 20                                                          INST 450
   80 READ (1,1001) DERIV(1),SMIN(1),SMAX(1)                            INST 460
   90 WRITE(3,2001)(CER(IDER(I),LATLON),I=1,NPLOT)                      INST 470
  100 PARM=PARAM                                                        INST 480
      WTPL=WTPLOT                                                       INST 490
      IF(NPARAM.GT.0) PARM=0.                                           INST 500
      DO 110 II=1,NBP                                                   INST 510
      IF((LONG(II).AND.(LATLON.EQ.1)) .OR.                              INST 520
     -     (.NOT.LONG(II).AND.(LATLON.EQ.2)))  GO TO 110                INST 530
      LL=II                                                             INST 540
      IF(PARM*BP(II)*(PARM-BP(II)).EQ.0.) GO TO 120                     INST 550
  110 CONTINUE                                                          INST 560
```

```
      WRITE(3,2003)                                                    INST 570
      WTPL=.FALSE.                                                     INST 580
 1001 FORMAT(4(A4,F6.0,F10.0))                                         INST 590
 2001 FORMAT(27H COEFFICIENTS TO BE PLOTTED/1X,2LA6)                   INST 600
 2002 FORMAT(1H0,A4,45H IS NOT A VALID DERIVATIVE NAME FOR THIS PLOT)  INST 610
 2003 FORMAT(30HONO WIND TUNNEL DATA AVAILABLE)                        INST 620
 2004 FORMAT(1H0,A4,6H PLOTS,5X,6HPARAM=,F10.4,5X,10HTOLERANCE=,F10.4) INST 630
  120 RETURN                                                           INST 640
      END                                                              INST 650
```

SUBROUTINE SUMPLT

Description: Subroutine SUMPLT plots data for one derivative.

Programing notes: Most of the data manipulation has been done, and the data are ready to plot. Thus this subroutine does little except the actual plotting.

Subroutine listing:

```
      SUBROUTINE SUMPLT(FDAT,FDATC,ALFS,ND2,DATA,NBP,NMBP,NABP)          SUMP   0
C     PLOTS SUMMARY INFORMATION FOR ONE DERIVATIVE                       SUMP  10
      COMMON /SUMDAT/ YLOC,XSKIP,ALEN,ASCAL2,YSTEP,AMIN,TABLAB,FDATA,    SUMP  20
     -     FDATAC,TITLE                                                 SUMP  30
      COMMON /PSCL/ CREACT,IWT1,IWT2,YLEN2,II,NPARAM                     SUMP  40
      COMMON /PDAT/ NOPLOT,YMN,YSCALE,NCAS,CRF,WTP,DERIV,WTO,KWT         SUMP  50
      COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,BP,NCMAX,LONG                    SUMP  60
      REAL FDAT(ND2,1),FDATC(ND2,1),ALFS(ND2,1),DATA(NBP,NMBP,1),        SUMP  70
     -     FDATA(5000),FDATAC(5000),TABLAB(2),ABP(16),MBP(16),BP(8),     SUMP  80
     -     TITLE(20),WTO(18,16),ABPS(18)                                 SUMP  90
      INTEGER NCAS(16),ISYMB(16)                                         SUMP 100
      LOGICAL NOPLOT,LONG(8),WTP                                         SUMP 110
      DATA ISYMB/1,0,5,2,12,10,6,9,4,11,7,8,A,B,8,8/                     SUMP 120
      NCMX=NCMAX+2                                                       SUMP 130
      NO=ND2*2                                                           SUMP 140
      CALL PSCALE(NCMX,NO,ND2,FDATA,FDATAC,FDAT,FDATC,ALFS,DATA,NBP,     SUMP 150
     -     NMBP,NABP)                                                    SUMP 160
      IF(II.NE.1) GO TO 10                                               SUMP 170
      CALL PLOT(0.,-YLOC,-3)                                             SUMP 180
      YLOC=0.                                                            SUMP 190
      CALL PLTDAT(XSKIP,12.25)                                           SUMP 200
   10 IF(NOPLOT) RETURN                                                  SUMP 210
      KWT1=KWT-1                                                         SUMP 220
      J=0                                                                SUMP 230
      DO 15 I=IWT1,IWT2                                                  SUMP 240
      J=J+1                                                              SUMP 250
   15 ABPS(J)=ABP(I)                                                     SUMP 260
      ABPS(KWT+1)=AMIN                                                   SUMP 270
      ABPS(KWT+2)=ASCAL2                                                 SUMP 280
      IF(YLOC.NE.0.) GO TO 30                                            SUMP 290
      CALL PLOT(XSKIP,YSTEP,-3)                                          SUMP 300
      CALL SYMBOL(0.,YSTEP,.14,TITLE(1),0.,4)                            SUMP 310
      DO 20 I=2,20                                                       SUMP 320
   20 CALL SYMBOL(999.,YSTEP,.14,TITLE(I),0.,4)                          SUMP 330
      YLOC=YSTEP                                                         SUMP 340
      GO TO 40                                                           SUMP 350
   30 CALL PLOT(0.,-YSTEP,-3)                                            SUMP 360
      YLOC=0.                                                            SUMP 370
   40 CALL AXIS(0.,0.,8HALPHA,-5,ALEN,0.,AMIN,ASCAL2)                    SUMP 380
      CALL AXIS(0.,0.,DERIV,4,YLEN2,90.,YMN,YSCALE)                      SUMP 390
      CALL SYMBOL(ALEN,YLEN2,.10,4HSYMB,0.,4)                            SUMP 400
      CALL SYMBOL(ALEN+.5,YLEN2,.10,TABLAB,0.,8)                         SUMP 410
      YORG=YLEN2                                                         SUMP 420
      DO 50 I=1,ND2                                                      SUMP 430
      IF(NCAS(I).LE.0) GO TO 50                                          SUMP 440
      YORG=YORG-.25                                                      SUMP 450
      CALL SYMBOL(ALEN+.1,YORG+.07,.14,ISYMB(I),0.,-1)                   SUMP 460
      BPVAL=MBP(I)                                                       SUMP 470
      IF(NPARAM.GT.0) BPVAL=BP(I)                                        SUMP 480
      CALL NUMBER(ALEN+.5,YORG,.14,BPVAL,0.,2)                           SUMP 490
   50 CONTINUE                                                           SUMP 500
C     PLOT FLIGHT POINTS                                                 SUMP 510
      DO  90 I=1,ND2                                                     SUMP 520
      NCI=NCAS(I)                                                        SUMP 530
      IF(NCI.LE.0) GO TO 70                                              SUMP 540
      ISI=ISYMB(I)                                                       SUMP 550
      DO 60 J=1,NCI                                                      SUMP 560
```

```
      XN=(ALFS(I,J)-AMIN)/ASCAL2                                    SUMP 570
      YN=(FDAT(I,J)-YMN)/YSCALE                                     SUMP 580
      CALL SYMBOL(XN,YN,.14,ISI,0.,-1)                              SUMP 590
      IF(FDATC(I,J).EQ.0.) GO TO 60                                 SUMP 600
      HITE=FDATC(I,J)/YSCALE                                        SUMP 610
      YNH=YN+HITE                                                   SUMP 620
      XNP=XN+.03                                                    SUMP 630
      XNM=XN-.03                                                    SUMP 640
      CALL PLOT(XNM,YNH,3)                                          SUMP 650
      CALL PLOT(XNP,YNH,2)                                          SUMP 660
      CALL PLOT(XN,YNH,3)                                           SUMP 670
      YNH=YN-HITE                                                   SUMP 680
      CALL PLOT(XN,YNH,2)                                           SUMP 690
      CALL PLOT(XNM,YNH,3)                                          SUMP 700
      CALL PLOT(XNP,YNH,2)                                          SUMP 710
   60 CONTINUE                                                      SUMP 720
C     PLOT WIND TUNNEL DATA                                         SUMP 730
   70 IF(.NOT.WTP) GO TO 90                                         SUMP 740
      IF(NPARAM.GT.0 .AND. I.EQ.1) GO TO 80                         SUMP 750
      IF(NPARAM.GT.0 .OR. NCI.EQ.0) GO TO 90                        SUMP 760
   80 WTO(KWT+1,I)=YMN                                              SUMP 770
      WTO(KWT+2,I)=YSCALE                                           SUMP 780
      CALL LINES(ARRS,WTO(1,I),KWT,1,KWT1,ISI)                      SUMP 790
   90 CONTINUE                                                      SUMP 800
      YN=-YMN/YSCALE                                                SUMP 810
      IF(YN.LE.0. .OR. YN.GT.YLEN2) GO TO 100                       SUMP 820
      CALL PLOT(ALEN,YN,3)                                          SUMP 830
      CALL PLOT(0.,YN,2)                                            SUMP 840
  100 CONTINUE                                                      SUMP 850
      RETURN                                                        SUMP 860
      END                                                           SUMP 870
```

SUBROUTINE PSCALE

Description: Subroutine PSCALE selects flight data points to be plotted on the basis of the criteria specified in subroutine INSTR. It places flight data and predicted derivatives for a single derivative into arrays for plotting and determines ordinate scales if needed.

Programing notes: Flight data are moved from arrays FDATA and FDATAC to arrays FDAT, FDATC, and ALFS. Array FDAT contains the derivative values, FDATC the confidence levels, and ALFS the angles of attack. Predicted derivatives are selected from array DATA and moved to array WTD.

Subroutine listing:

```
      SUBROUTINE PSCALE(NCMX,ND,ND2,FDATA,FDATAC,FDAT,FDATC,ALFS,DATA,   PSCA   0
     -     NBP,NMBP,NAPP)                                                PSCA  10
C     DETERMINES PLOT SCALES, SELECTS DATA TO BE PLOTTED                 PSCA  20
C     DATA TO BE PLOTTED IS SELECTED FROM ARRAYS FDATA AND FDATAC        PSCA  30
C          AND PLACED INTO THE SMALLER ARRAYS FDAT,FDATC, AND ALFS       PSCA  40
      COMMON /CASES/ NCASE                                               PSCA  50
      COMMON /PSCL/ CRFACT,IWT1,IWT2,YLEN2,II,NPARAM                     PSCA  60
      COMMON /SELECT/ PARAM,TOL,IDER,YMIN,YMAX,LATLON,DERIVS,LL,WTPL     PSCA  70
      COMMON /PDAT/ NOPLOT,YMN,YSCALE,NCAS,CRF,WTP,DERIV,WTC,KWT         PSCA  80
      REAL FDATA(NCMX,ND,1),FDATAC(NCMX,ND,1),FDAT(ND2,1),FDATC(ND2,1),  PSCA  90
     -     ALFS(ND2,1),DATA(NBP,NMBP,1),YMIN(21),YMAX(21),ZSC(4),        PSCA 100
     -     DERIVS(21),WTD(18,16)                                         PSCA 110
      INTEGER IDER(21),NCAS(16),NCASE(32)                                PSCA 120
      LOGICAL WTPL,WTP,NOPLOT                                            PSCA 130
      WTP=WTPL                                                           PSCA 140
      LONLAT=ND2*(2-LATLON)                                              PSCA 150
      JDER=IDER(II)                                                      PSCA 160
      DERIV=DERIVS(II)                                                   PSCA 170
      ZSC(1)=0.                                                          PSCA 180
      ZSC(2)=0.                                                          PSCA 190
      CRF=CRFACT                                                         PSCA 200
      IF(JDER.GT.19) CRF=0.                                              PSCA 210
      IF(JDER.EQ.21) WTP=.FALSE.                                         PSCA 220
      NOPLOT=.TRUE.                                                      PSCA 230
      DO 90 JJ=1,ND2                                                     PSCA 240
      JJL=JJ+LONLAT                                                      PSCA 250
      NCJ=0                                                              PSCA 260
      NCASEJ=NCASE(JJL)                                                  PSCA 270
      IF(NCASEJ.LE.0) GO TO 60                                           PSCA 280
      DO 50 I=1,NCASEJ                                                   PSCA 290
      IF( PARAM*FDATAC(22,JJL,I).NE.0. .AND.                             PSCA 300
     -     ABS(PARAM-FDATAC(22,JJL,I)).GT.TOL)  GO TO 50                 PSCA 310
      IF(FDATAC(JDER,JJL,I).LE.0.) GO TO 50                              PSCA 320
      NCJ=NCJ+1                                                          PSCA 330
      FDAT(JJ,NCJ)=FDATA(JDER,JJL,I)                                     PSCA 340
      FDATC(JJ,NCJ)=FDATAC(JDER,JJL,I)*CRF                               PSCA 350
      ALFS(JJ,NCJ)=FDATA(22,JJL,I)                                       PSCA 360
      ZSC(1)=AMIN1(ZSC(1),FDAT(JJ,NCJ)-FDATC(JJ,NCJ))                    PSCA 370
      ZSC(2)=AMAX1(ZSC(2),FDAT(JJ,NCJ)+FDATC(JJ,NCJ))                    PSCA 380
   50 CONTINUE                                                           PSCA 390
   60 IF(.NOT. WTP) GO TO 85                                             PSCA 400
      IF(NPARAM.GT.0 .AND. JJ.EQ.1) GO TO 70                             PSCA 410
      IF(NPARAM.GT.0 .OR. NCJ.EQ.0) GO TO 85                             PSCA 420
   70 K1=(JDER-1)*NAPP+IWT1                                              PSCA 430
      KWT=IWT2-IWT1+1                                                    PSCA 440
      K2=K1+IWT2-IWT1                                                    PSCA 450
      J=0                                                                PSCA 460
      DO 80 I=K1,K2                                                      PSCA 470
      J=J+1                                                              PSCA 480
      WTD(J,JJ)=DATA(LL,JJ,I)                                            PSCA 490
      ZSC(1)=AMIN1(ZSC(1),DATA(LL,JJ,I))                                 PSCA 500
   80 ZSC(2)=AMAX1(ZSC(2),DATA(LL,JJ,I))                                 PSCA 510
   85 NCAS(JJ)=NCJ                                                       PSCA 520
   90 NOPLOT=NOPLOT .AND. (NCJ.EQ.0)                                     PSCA 530
      IF(NOPLOT) GO TO 110                                               PSCA 540
      IF(YMAX(II).EQ.YMIN(II)) GO TO 100                                 PSCA 550
      YMN=YMIN(II)                                                       PSCA 560
```

```
      YSCALE=(YMAX(II)-YMIN(II))/YLEN2                                  PSCA 570
      GO TO 200                                                         PSCA 580
  100 CALL SCALES(ZSC,YLEN2,2,.FALSE.)                                  PSCA 590
      YMN=ZSC(3)                                                        PSCA 600
      YSCALE=ZSC(4)                                                     PSCA 610
      GO TO 200                                                         PSCA 620
  110 WRITE(3,2000)DERIV                                                PSCA 630
 2000 FORMAT(30HNO FLIGHT DATA AVAILABLE FOR ,A4)                       PSCA 640
  200 RETURN                                                            PSCA 650
      END                                                               PSCA 660
```

SUBROUTINES WINDIN, LOAD1, SCALES, LINES, PLTDAT, TIME, AND DATE

Subroutines WINDIN, LOAD1, SCALES, LINES, PLTDAT, TIME, and DATE are identical to those in the SETUP and MMLE program.

APPENDIX F

SAMPLE CASE FOR THE SUMARY PROGRAM

This appendix presents a sample case for the SUMARY program.

INPUT CARDS

```
SAMPLE CASE FOR SUMARY
 $WIND NCLO=6,LONG(1)=T,NARP=4,BODY=T,AMAX=24.,ASCALE=2.,CREACT=10.,$END
CN        2
 .4          .65        .9         1.
CNA       2
 .075        .07        .065       .06
CMA       2
 -.006       -.005      -.008      -.012
CNDE      1
 .02
CMDE      1
 -.01
CMQ       2
 -5.         -5.         -5.2      -6.
 5.          10.        15.        20.
 .5
 1.
LONG   AIRCRAFT B     FLT 1 CASE 1              0.000      4.803      1.000       .260
A          3          4
  -.069736  0.000000  0.000000  -.005420
  -.005095 -5.276583  0.000000  0.000000
   .003742  0.000000  0.000000  0.000000
B          3          6
  -.008488  0.000000  0.000000  0.000000  -.346488   .076020
  -.010056  0.000000  0.000000  0.000000 -2.288757  0.000000
   .003654  0.000000  0.000000  0.000000  -.328265  0.000000
AC         3          3
   .000720  0.000000  0.000000
   .000077   .362775  0.000000
  0.000000  0.000000  0.000000
BC         3          5
   .000656  0.000000  0.000000  0.000000   .004551
   .000213  0.000000  0.000000  0.000000   .000825
  0.000000  0.000000  0.000000  0.000000  0.000000
LONG   AIRCRAFT B     FLT 1  CASE 4             0.000     10.031      1.000       .260
A          3          4
  -.068788  0.000000  0.000000  -.016320
  -.004044 -6.148364  0.000000  0.000000
   .006651  0.000000  0.000000  0.000000
B          3          6
  -.006378  0.000000  0.000000  0.000000  -.653271   .115600
  -.010248  0.000000  0.000000  0.000000 -5.328408  0.000000
   .002856  0.000000  0.000000  0.000000  -.011942  0.000000
AC         3          3
   .001169  0.000000  0.000000
   .000086   .403856  0.000000
  0.000000  0.000000  0.000000
BC         3          5
   .001016  0.000000  0.000000  0.000000   .013379
   .000192  0.000000  0.000000  0.000000   .001911
  0.000000  0.000000  0.000000  0.000000  0.000000
LONG   AIRCRAFT B     FLT 1  CASE 8              0.000     13.671      1.000       .260
A          3          4
  -.051724  0.000000  0.000000  -.026070
  -.005240 -2.218513  0.000000  0.000000
   .005995  0.000000  0.000000  0.000000
B          3          6
  -.019702  0.000000  0.000000  0.000000  -.813778   .112380
  -.009046  0.000000  0.000000  0.000000 -6.745284 -0.000000
   .002492  0.000000  0.000000  0.000000  -.002600  0.000000
AC         3          3
   .001280  0.000000  0.000000
   .000044   .296974  0.000000
  0.000000  0.000000  0.000000
BC         3          5
   .001391  0.000000  0.000000  0.000000   .020161
   .000202  0.000000  0.000000  0.000000   .001633
  0.000000  0.000000  0.000000  0.000000  0.000000
```

```
LONG   AIRCRAFT B    FLT 1   CASE 11          0.000     16.399      1.000       .260
A         3         4
  -.058530  0.000000  0.000000  -.012960
  -.008326 -2.095303  0.000000  0.000000
   .005033  0.000000  0.000000  0.000000
B         3         6
  -.013062  0.000000  0.000000  0.000000  -.935426   .120320
  -.010224  0.000000   .000046  0.000000 -8.717713 -0.000000
   .002516  0.000000  0.000000  0.000000   .002881  0.000000
AC        3         3
   .001301  0.000000  0.000000
   .000051   .219645  0.000000
  0.000000  0.000000  0.000000
BC        3         5
   .001209  0.000000  0.000000  0.000000   .023573
   .000147  0.000000  0.000000  0.000000   .001681
  0.000000  0.000000  0.000000  0.000000  0.000000
LONG   AIRCRAFT B    FLT 1   CASE 12          0.000     17.993      1.000       .260
A         3         4
  -.066610  0.000000  0.000000  -.015460
  -.007499 -4.851715  0.000000  0.000000
   .005784  0.000000  0.000000  0.000000
B         3         6
   .001381  0.000000  0.000000  0.000000 -1.008667   .171330
  -.011126  0.000000  0.000000  0.000000 -9.911981 -0.000000
   .002070  0.000000  0.000000  0.000000   .004800  0.000000
AC        3         3
   .001335  0.000000  0.000000
   .000056   .222093  0.000000
  0.000000  0.000000  0.000000
BC        3         5
   .001252  0.000000  0.000000  0.000000   .027691
   .000157  0.000000  0.000000  0.000000   .002271
  0.000000  0.000000  0.000000  0.000000  0.000000
LONG   AIRCRAFT B    FLT 1   CASE 15          0.000     20.088      1.000       .260
A         3         4
  -.061936  0.000000  0.000000  -.003870
  -.010069 -3.778632  0.000000  0.000000
   .006358  0.000000  0.000000  0.000000
B         3         6
  -.006655  0.000000  0.000000  0.000000 -1.085620   .148190
  -.011583  0.000000  0.000000  0.000000-11.977410 -0.000000
   .001551  0.000000  0.000000  0.000000   .012342  0.000000
AC        3         3
   .001470  0.000000  0.000000
   .000073   .254969  0.000000
  0.000000  0.000000  0.000000
BC        3         5
   .001046  0.000000  0.000000  0.000000   .030336
   .000171  0.000000  0.000000  0.000000   .002594
  0.000000  0.000000  0.000000  0.000000  0.000000
LONG   AIRCRAFT B    FLT 1   CASE 17          0.000     21.376      1.000       .260
A         3         4
  -.055533  0.000000  0.000000  -.008290
  -.011031  -.436157  0.000000  0.000000
   .006307  0.000000  0.000000  0.000000
B         3         6
  -.015876  0.000000  0.000000  0.000000 -1.154961   .164710
  -.008337  0.000000  0.000000  0.000000-13.301825 -0.000000
   .001350  0.000000  0.000000  0.000000   .011709  0.000000
AC        3         3
   .001023  0.000000  0.000000
   .000039   .133743  0.000000
  0.000000  0.000000  0.000000
BC        3         5
   .000712  0.000000  0.000000  0.000000   .021764
   .000086  0.000000  0.000000  0.000000   .001495
  0.000000  0.000000  0.000000  0.000000  0.000000
```

```
LONG   AIRCRAFT B     FLT 1   CASE 19          0.000     22.641     1.000       .260
A         3          4
  -.057218  0.000000  0.000000  -.005510
  -.011844   .165115  0.000000  0.000000
   .006115  0.000000  0.000000  0.000000
B         3          6
  -.012764  0.000000  0.000000  0.000000 -1.187400    .185320
  -.007367  0.000000  0.000000  0.000000-15.322314 -0.000000
   .001150  0.000000  0.000000  0.000000    .012945  0.000000
AC        3          3
   .001052  0.000000  0.000000
   .000060   .161152  0.000000
  0.000000  0.000000  0.000000
BC        3          5
   .000772  0.000000  0.000000  0.000000    .025420
   .000118  0.000000  0.000000  0.000000    .002325
  0.000000  0.000000  0.000000  0.000000  0.000000
PLOT
LATR
CNB
LONG
CN                    CNA                   CMDE                  CNDE
CMA                   CMQ                   DE                    CMDC
END
```

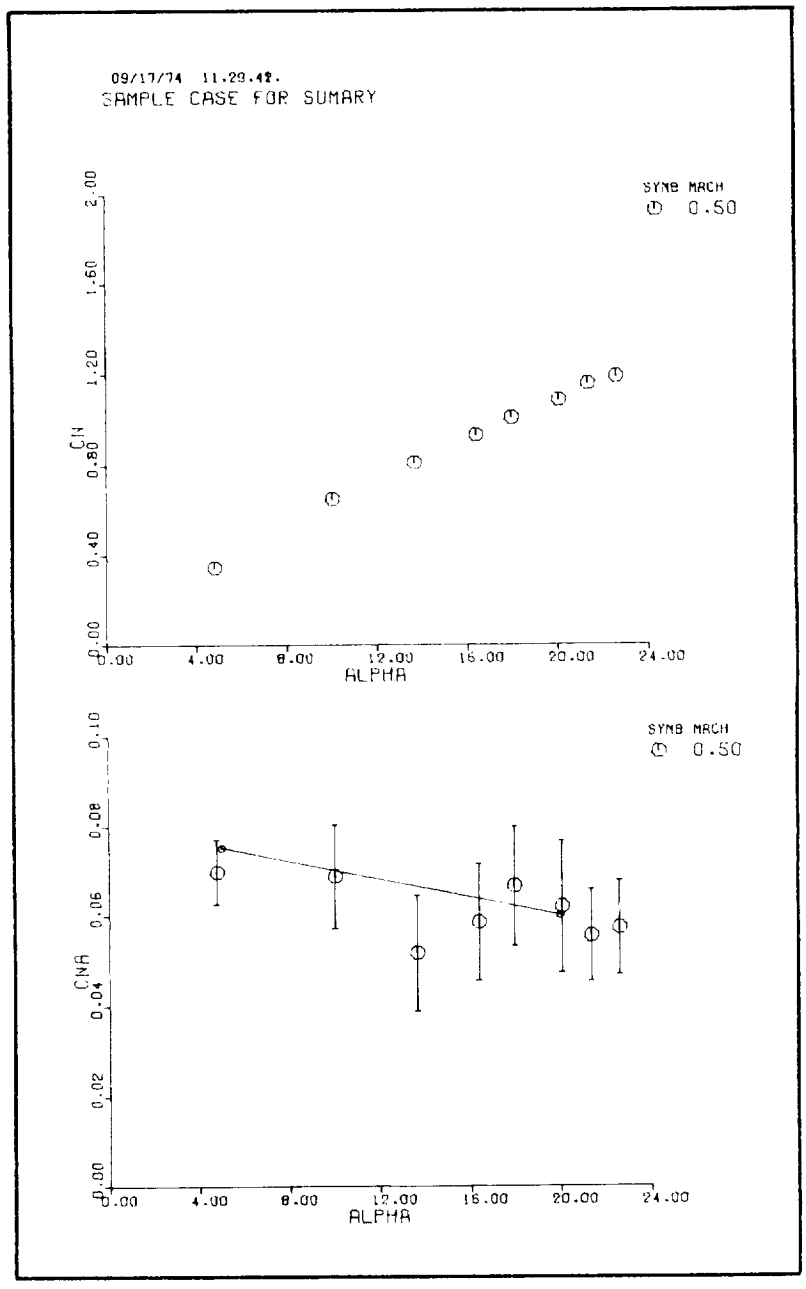
OUTPUT LISTING

```
MMLE SUMMARY PLOTTING PROGRAM     ****   1 JULY 1974     ****     VERSION 2

                    SAMPLE CASE FOR SUMARY

CONFIDENCE LEVELS WILL BE PLOTTED MULTIPLIED BY 10.0

LONG          AIRCRAFT B     FLT 1 CASE 1          0.0000    4.8030    1.0000    .2600

LONG          AIRCRAFT B     FLT 1  CASE 4          0.0000   10.0310    1.0000    .2600

LONG          AIRCRAFT B     FLT 1  CASE 8          0.0000   13.6710    1.0000    .2600

LONG          AIRCRAFT B     FLT 1   CASE 11         0.0000   16.3990    1.0000    .2600

LONG          AIRCRAFT B     FLT 1   CASE 12         0.0000   17.9930    1.0000    .2600

LONG          AIRCRAFT B     FLT 1   CASE 15         0.0000   20.0880    1.0000    .2600

LONG          AIRCRAFT B     FLT 1   CASE 17         0.0000   21.3760    1.0000    .2600

LONG          AIRCRAFT B     FLT 1   CASE 19         0.0000   22.6410    1.0000    .2600

LATR PLOTS      PARAM=   -0.0000      TOLERANCE=   -0.0000
COEFFICIENTS TO BE PLOTTED
CNB

NO WIND TUNNEL DATA AVAILABLE

NO FLIGHT DATA AVAILABLE FOR CNB

LONG PLOTS      PARAM=   -0.0000      TOLERANCE=   -0.0000
COEFFICIENTS TO BE PLOTTED
CN    CNA   CMDE  CNDE  CMA   CMQ   DE    CMDC

NO FLIGHT DATA AVAILABLE FOR CMDC
```

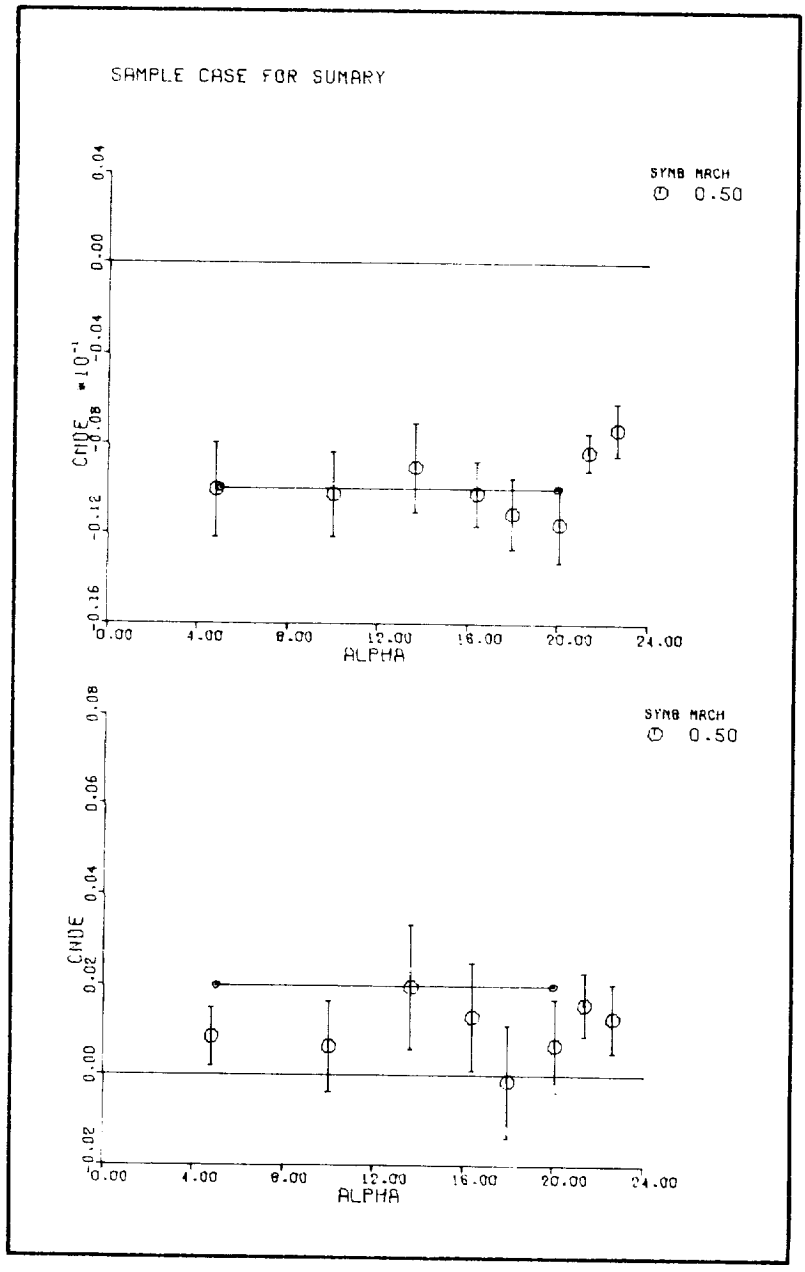
A sample plot from the SUMARY program is shown. The plot is presented in four parts to avoid loss of detail from a large reduction. The plot as produced by the automatic plotter is shown within the heavy lines. Explanatory material is included to aid the user in implementing the program. Solid lines denote predicted derivatives. Vertical bars (I) indicate confidence levels.

C_N

C_{N_α}



α

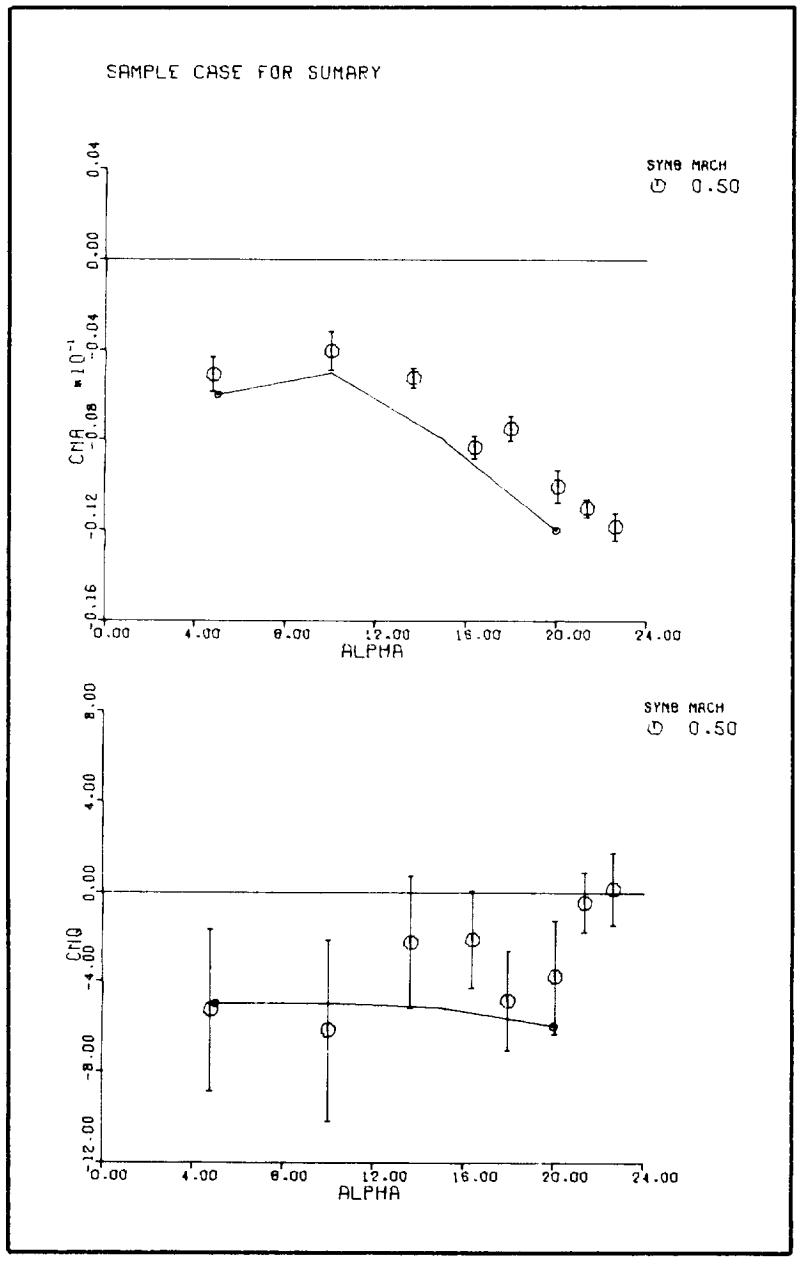


$C_{m_{\delta_e}}$

$C_{N_{\delta_e}}$

α

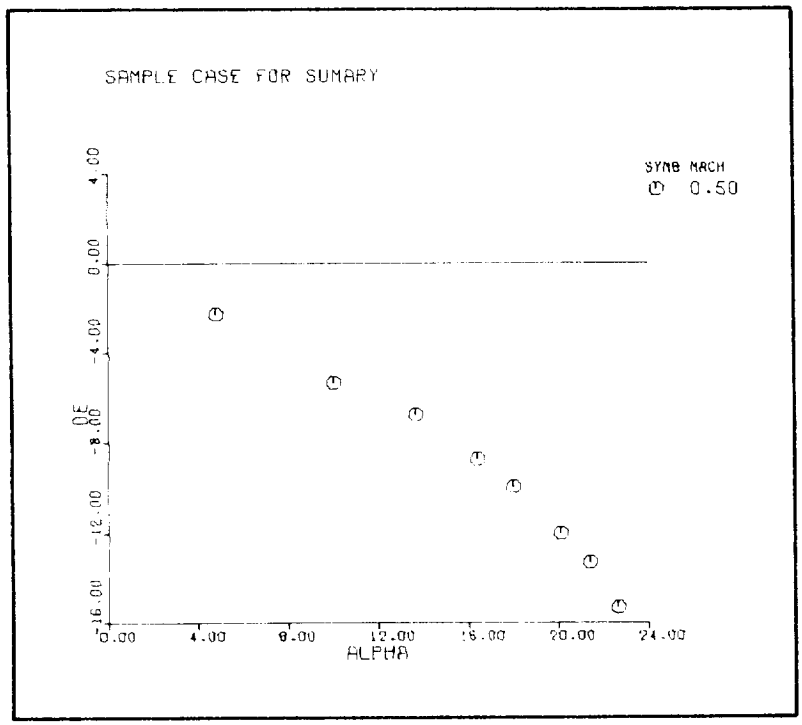
C_{m_α}



C_{m_q}

α

$\delta_{e_{trim}}$



α

REFERENCES

1. Taylor, Lawrence W., Jr.; and Iliff, Kenneth W.: Systems Identification Using a Modified Newton-Raphson Method – A FORTRAN Program. NASA TN D-6734, 1972.

2. Iliff, K. W.: Identification and Stochastic Control With Application to Flight Control in Turbulence. UCLA-ENG-7340, School of Engineering and Applied Science, Univ. Calif., Los Angeles, Calif., May 1973.

3. Iliff, Kenneth W.; and Taylor, Lawrence W., Jr.: Determination of Stability Derivatives From Flight Data Using a Newton-Raphson Minimization Technique. NASA TN D-6579, 1972.

4. Balakrishnan, A. V.: Communication Theory. McGraw-Hill Book Co., Inc., c.1968.

5. Anon.: Control Data Cyber 70 Series Computer Systems – 6000 Series Computer Systems – 7600 Computer System. LOADER Reference Manual. Control Data Corp., c.1974.

6. Anon.: Programming CalComp Pen Plotters. California Computer Products, Inc., Sept. 1969.

7. Ogata, Katsuhiko: State Space Analysis of Control Systems. Prentice-Hall, Inc., c.1967.

8. Iliff, Kenneth W.; and Maine, Richard E.: Practical Aspects of Using a Maximum Likelihood Estimator. NASA paper presented at AGARD Flight Mechanics Panel Specialists Meeting, Hampton, Va., Nov. 5-8, 1974.

9. Noble, Ben: Applied Linear Algebra. Prentice-Hall, Inc., c.1969.